

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR



FACULTAD DE INGENIERÍA

MAESTRÍA EN REDES DE COMUNICACIÓN

INFORME FINAL CASO DE ESTUDIO PARA UNIDAD DE TITULACIÓN
ESPECIAL

TEMA:

**“Monitoreo Ambiental aplicado a un mini invernadero usando tecnología
ZigBee e INTERNET”**

Mario Santiago López Montenegro

Quito – 2016

Contenido

1.	Introducción	1
2.	Justificación	2
3.	Antecedentes	3
4.	Objetivos	4
4.1	Objetivo general.....	4
4.2	Objetivos Específicos:	4
4	Desarrollo del caso de Estudio: Estudio comparativo de sistemas de monitoreo para invernaderos	4
5.1	Sistema de monitoreo AIRPI	5
4.2	Sistema de Monitoreo Weather Meters.....	6
5.3	Sistema de Monitoreo GrovePi.....	7
5.4	Ventajas.....	10
5.5	Desventajas	10
6	Configurar los parámetros de los componentes para la red ZigBee y la conexión a Internet.....	11
6.1	Configuración de los módulos Xbee.....	11
6.2	Configuración del módulo Coordinador	11
6.3	Configuración del módulo Router.....	14
6.4	Configuración de módulos Xbee con el programa XCTU	16
7	Elaboración del Prototipo	20
7.1	Implementación del sensor de luz	22
7.2	Implementación del sensor de temperatura	25
7.4	Armado del sistema de gestión del Mini invernadero.....	32
7.5	Puesta en marcha del Sistema de monitoreo ambiental.....	34
8	Pruebas de funcionamiento del prototipo.....	37
8.1	Pruebas de Laboratorio.....	37
8.2	Pruebas de campo.....	39
8.2.1	Pruebas con obstáculos	39
8.2.2	Pruebas con línea de Vista	42
9	Comparación del prototipo con otros sistemas de monitoreo.....	47
9.1	Ventajas del Prototipo de monitoreo Ambiental con otros sistemas de monitoreo	49
9.2	Desventajas del Prototipo de monitoreo Ambiental con otros sistemas de monitoreo	49
10.	Conclusiones y Recomendaciones	50

10.1 Conclusiones	50
10.2 Recomendaciones	50
11 Bibliografía	51
12 ANEXOS	53

Ilustraciones

Figura 1: AirPi	5
Figura 2: Weather Meters	6
Figura 3: GrovePi	7
Figura 4: Configuración Xbee	11
Figura 5: Dirección DH y DL del Gateway Xbee	12
Figura 6: Configuración Básica del Módulo Xbee Coordinador	12
Figura 7: Campo PAN ID	13
Figura 8: Calanes disponibles para selección	13
Figura 9: Distribución de Canales	13
Figura 10: Configuración Básica del Módulo Xbee Router	14
Figura 11: Selección de Canales	14
Figura 12: Configuración DH/DL extendida	15
Figura 13: Programador con Módulo Xbee S2 Pro	16
Figura 14: Aplicativo XCTU	17
Figura 15: Parámetros para la comunicación serial	17
Figura 16: Dirección DH y DL impresa en el módulo Xbee	18
Figura 17: Conexión del módulo Xbee en XCTU	18
Figura 18 : Configuración para la conexión punto a punto básica	19
Figura 19: Configuración básica para la conexión con el coordinador directamente	19
Figura 20: Configuración básica para la conexión punto a punto con cualquier otro nodo	20
Figura 21: Configuración pines analógicos y digitales	20
Figura 22: Fotocelda	21
Figura 23: Sensor de Temperatura	21
Figura 24: Módulo Xbee Pro S2	22
Figura 25: Gateway	22
Figura 26: Configuración de Pines	23
Figura 27: Esquema del circuito de la fotocelda	23
Figura 28: Circuito de la Fotocelda	24
Figura 29: Parametrización del Widget	25
Figura 30: Widget para la medición de luminosidad	25
Figura 31: Configuración de Pines Analógicos	26
Figura 32: Diseño del circuito para el sensor de temperatura	27
Figura 33: Conexión del sensor de Temperatura	27
Figura 34: Conexión de las resistencias	28
Figura 35: Creación del Widget de temperatura	29
Figura 36: Dashboard para el monitoreo del invernadero	29

Figura 37: Creación del Widget del módulo de Mensajes	30
Figura 38: Dashboard para el monitoreo Ambiental	31
Figura 39: Equipos para la comunicación y monitoreo de invernadero	33
Figura 40: Parámetros para la conexión serial hacia el módulo Xbee	34
Figura 41: Sensores y Xbee Router dentro del invernadero	35
Figura 42: Esquema general del monitoreo del invernadero por ZigBee e Internet	35
Figura 43: Esquema general del monitoreo del invernadero por ZigBee e Internet	36
Figura 44: Pantalla de monitoreo del invernadero	37
Figura 45: Leds informativos Gateway	37
Figura 46: Prototipo dentro del invernadero	38
Figura 47: Browser mostrando los datos de sensores	38
Figura 48: Envío de mensajes de Texto	39
Figura 49: Resultado de las pruebas a 10m	40
Figura 50: Resultado de las pruebas a 20m	40
Figura 51: Resultado de las pruebas a 30m	41
Figura 52: Resultado de las pruebas a 50m	41
Figura 53: Gateway energizada por Baterías	42
Figura 54: Ubicación para las pruebas a 50m	43
Figura 55: Resultado de las pruebas a 50m	43
Figura 56: Ubicación para las pruebas a 100m	44
Figura 57: Ubicación para las pruebas a 100m	44
Figura 58: Ubicación para las pruebas a 150m	45
Figura 59: Resultado de las pruebas a 150m	45
Figura 60: Ubicación para las pruebas a 200m	46
Figura 61: Resultado de las pruebas a 200m	46

Tablas

Tabla 1: Comparativo de Sistemas de monitoreo Ambiental	9
Tabla 2: Resultado de Pruebas con Obstáculos	42
Tabla 3: Resultado de Pruebas sin Obstáculos	47
Tabla 4: Comparativo del Prototipo con los Sistemas de Monitoreo Ambiental	48

1. Introducción

Este caso de estudio pretende construir un prototipo de monitoreo ambiental de un mini invernadero basado en la tecnología ZigBee utilizando módulos Xbee, sensores y un GateWay que integrará la red ZigBee con Internet para la visualización de variables climáticas a través de un browser el que mostrará la temperatura y la luminosidad del ambiente dentro del mini invernadero y también en la implementación de un pequeño módulo de envío de mensajes de texto a través de una consola y un módulo HTML.

El funcionamiento del prototipo consiste en un módulo coordinador el cual es el que recibe información de un módulo sensor este módulo sensor es un dispositivo Xbee donde se encuentran conectados los sensores de luz y de temperatura, este módulo Xbee se situará dentro del mini invernadero junto con los sensores climáticos para el registro de las variables estos datos se envían al coordinador y a través del GateWay este a su vez envía estos datos al internet para ser interpretados por un Browser.

El módulo de envío de mensajes de texto es un módulo extra que sirve para comunicar cualquier decisión o enviar alguna orden remota para ser leída en el sitio del invernadero y de ese modo demostrar que el prototipo no solo es exclusivo para conexión de sensores sino para otras funciones.

Administrar una red de control de invernaderos puede ser muy costoso por su monitoreo constante, su implementación, soporte técnico y administración para mantener las condiciones óptimas de los cultivos. Es por eso que cultivadores de plantas de invernadero recurren a la tecnología para monitorear y automatizar sus cultivos. El uso de sensores de temperatura, humedad, pH del suelo, luminosidad, etc. en los invernaderos, proporcionan datos importantes para la gestión de los mismos. Sin embargo, no todos los cultivadores pueden darse el lujo de tener un sistema de monitoreo de invernaderos ya que aquellos que se encuentran en el mercado aún poseen altos costos para su implementación, gestión y soporte.

Cabe mencionar que, en el mercado la mayoría de sistemas de monitoreo existentes manejan sensores en una sola estación independiente, es decir que, no existe gran diversidad de proyectos de monitoreo ambiental basados en software libre o algún tipo de sistema de monitoreo que utilice la tecnología ZigBee, con software libre y de bajo costo. Las ventajas que proporciona la tecnología ZigBee son: El manejo de mayores distancias y que los nodulos Xbee pueden interactuar entre sí para manejar un conjunto de módulos Xbee como un solo sistema de monitoreo.

En este contexto, se planteara el siguiente caso de estudio, que consiste en la creación de un sistema de monitoreo ambiental para cualquier tipo de necesidad. Este sistema de monitoreo sirve para monitorear cualquier tipo de ambiente, ya sea interiores o exteriores, además de ser *escalable hasta 65535 nodos a una distancia de 1 milla entre nodos (con módulos Xbee S2 Pro)*. Para ello, se pondrá como ejemplo el monitoreo de un mini invernadero en el cual se medirán dos variables: luminosidad, temperatura de forma automática y un pequeño sistema de mensajes de texto. Es necesario mencionar que, los datos de las variables ambientales se publicará en un *Browser* desde cualquier parte del mundo con acceso a *Internet*.

Gracias a la tecnología ZigBee, los dispositivos XBee, algunos sensores, un Gateway y conexión a Internet, se permite obtener los registros de medición proporcionados directamente en un browser, a un bajo costo, para ser mostrados de una forma amigable, simple de administrar y fácil de escalar, y posteriormente tomar decisiones oportunas.

2. Justificación

El caso de estudio busca crear un prototipo para el monitoreo ambiental de bajo costo y con elementos de bajo consumo energético, fácil de administrar, fácil de escalar y fácil de monitorear aprovechando las ventajas del estándar ZigBee y como otra alternativa a los sistema de monitoreo ambiental que existen en el mercado.

El monitoreo ambiental aplicándolo a un mini invernadero nos sirve de guía para el mediano y pequeño agricultor, brindándonos parámetros para la implementación de un sistema de control climático automatizado, fácil de implementar, fácil de escalar y a un bajo costo, de manera que este sistema se adecue a cualquier tipo de cultivo de invernadero.

Controlar las variables climáticas es muy importante ya que hasta las más mínimas variaciones el ph (grado de acidez o basicidad) o ec (conductividad eléctrica) del agua, la temperatura y la humedad, incluso los niveles de iluminación, pueden impactar el crecimiento, salud de las plantas y producción de los cultivos. No obstante, el monitoreo ambiental abarca un sinnúmero de campos dependiendo de la utilidad que se necesite cubrir, por ejemplo, el monitoreo en zonas de desastres, mediante el cual se pueden prevenir accidentes y crear planes de contingencia y evacuaciones oportunas. Asimismo, en centros de datos este sistema ayuda a prolongar la vida de los equipos y la seguridad de la información, e incluso se puede aplicar en ciudades con gran cantidad de contaminación ya que se puede monitorear los niveles de CO2.

3. Antecedentes

El monitoreo de las variables climáticas tales como la temperatura, humedad, presión atmosférica, altitud, entre otras, son de gran importancia independientemente del campo de aplicación. Es decir que, el monitoreo ambiental describe el estado del medio ambiente y el posible comportamiento de variables climáticas, para la toma de acciones orientadas a la solución de problemas y planes de contingencia, así como tener datos estadísticos que nos sirven de guía para diagnosticar un posible comportamiento del medio ambiente en un futuro.

Ya sea en invernaderos, data centers y hospitales, zonas de riesgo, alta mar, etc., los sistemas de monitoreo ambiental nos permiten tener una idea cualitativa y cuantitativa de las variables climáticas para poder tomar decisiones acertadas (Meléndez & Cano, 2015, pág. 1).

Debido a que los cultivos bajo invernadero son una técnica reciente orientada a mejorar la producción de los cultivos, para protegerlos de los factores climáticos, controlar su crecimiento, controlar las plagas, crear micro climas, etc., (Lorenzo, 2012, pág. 24), este caso de estudio propone crear un sistema para el monitoreo de invernaderos de bajo costo, fácil de implementar, de fácil soporte técnico y fácil de escalar y de esta manera, agilizar la toma de decisiones según los datos proporcionados por sensores, a través de la tecnología ZigBee e Internet.

El monitoreo de invernadero proporciona los datos para el control interno del invernadero para así tomar decisiones acertadas para mejorar producción de cultivo de frutas, flores, plantas exóticas, hortalizas, etc. Las ventajas del sistema de invernaderos automatizados es la mayor productividad por m²; la garantía de tener una producción de calidad; el control eficiente de plagas y enfermedades del cultivo; un mayor control de los factores ambientales para poder producir fuera de época; tener las condiciones ambientales para producir cultivos rentables y tener más oportunidad de comercializar cultivos de alta calidad en un mercado competitivo.

Por ello, muchos cultivadores optan por la opción de automatizar sus invernaderos existiendo en el mercado varias empresas dedicadas a este propósito. En los últimos años son muchos los agricultores que han iniciado la instalación de sistemas que permiten la automatización ventilación, sensores que indican el grado de luminosidad en el interior, instalación de equipos de calefacción, etc. Sin embargo, para un mediano productor estas soluciones son demasiado costosas.

4. Objetivos

4.1 Objetivo General

Crear un Prototipo de Monitoreo Ambiental con tecnología ZigBee e implementarlo en un mini invernadero.

4.2 Objetivos Específicos:

1. Estudio comparativo de sistemas de monitoreo para invernaderos.
2. Configurar los parámetros de los componentes para la red ZigBee y la conexión a Internet.
3. Elaboración del prototipo.
4. Pruebas de funcionamiento del prototipo.
5. Comparación del prototipo con otros sistemas de monitoreo
6. Realizar un artículo referente al Monitoreo Ambiental aplicado a un mini invernadero usando tecnología ZigBee e Internet.

4 Desarrollo del caso de Estudio: Estudio comparativo de sistemas de monitoreo para invernaderos

Para este análisis presentará tres proyectos similares a al propuesto en este caso de estudio, es decir proyectos que utilicen software libre y componentes de bajo costo y de bajo consumo energético como son el AIR PI, Weather Methers y GrovePi con el fin de compararlos con este proyecto y definir sus ventajas y desventajas frente a la tecnología ZigBee. Los proyectos a comparar son multipropósito es decir nos sirven para monitorear ya sea el medio ambiente de un hogar.

Cabe mencionar que, en el mercado no hay muchos proyectos basados en software libre y con tecnología ZigBee, por lo cual se hará la comparación con los proyectos más parecidos al desarrollado en este caso de estudio.

En este capítulo se mencionaran las características más relevantes de estos sistemas de monitoreo ambiental y se los comparará con el proyecto del caso de estudio.

5.1 Sistema de monitoreo AIRPI

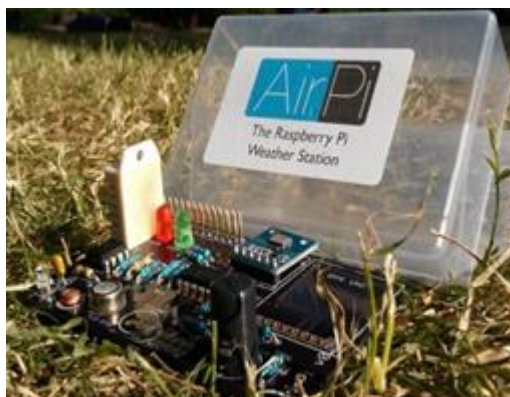


Figura 1: AirPi
Fuente: AirPi¹

El AirPi es un proyecto orientado al monitoreo del clima y la calidad del aire, usando como base la conectividad a un Raspberry Pi. El AirPi está conectado a varios sensores y tiene su propio software con código abierto que permite transformar los datos proporcionados por los sensores en datos útiles para el usuario, para luego ser publicados directamente en internet.

El AirPi físicamente es una placa que contiene un conjunto de sensores que se incorpora sensor de temperatura, humedad, presión del aire, los niveles de dióxido de nitrógeno, monóxido de carbono y los niveles de contaminación. El costo aproximado del AirPi es \$90 y no incluye la Raspberry Pi, pero si se quiere disminuir los costos se puede ordenar no la versión completa sino solo los sensores que se requieran.

Sus creadores aseguran que el dispositivo puede correr por meses autónomamente y que el costo de mantenimiento es realmente bajo. Es un dispositivo portátil y puede funcionar a altas temperaturas así como bajo cero grados

El AirPi sirve a personas interesadas en el monitoreo de la calidad del aire donde viven en grandes ciudades o sitios remotos que no son monitoreados. Los ambientalistas podrían sacar un gran provecho a este tipo de dispositivos al medir la calidad del aire cerca de minas o áreas de producción industrial.

Con AirPi, se puede controlar la calidad del aire en un ambiente cerrado ya sea su casa u oficina así también obtener estadísticas por ejemplo a qué hora del día en que el aire es más limpio.

¹ Revisar página web: <http://airpi.es/>

AirPi puede ser un recurso muy valioso para las personas que viven en sitios donde no hay registros de información meteorológica, como pueblos o tribus remotas. Se puede utilizar para predecir la aparición de mal tiempo usando datos como el cambio de la presión del aire y la humedad; debido a su capacidad de cargar automáticamente a Internet, una AirPi colocado correctamente puede proporcionar datos meteorológicos de fácil acceso para toda la comunidad.

Detalles:

- Sensor de CO2 y de NO2
- DHT22 Sensor de Temperatura y Humedad
- Sensor de luz LDR
- Micrófono para medir la contaminación acústica

4.2 Sistema de Monitoreo Weather Meters



Figura 2: Weather Meters
Fuente: Sparfun²

Es un proyecto enfocado al agricultor, al meteorólogo o simplemente a las personas interesadas al monitoreo ambiental, muy útil es esta estación metrológica que permite medir la velocidad del viento, su dirección y la detección lluvia.

Los sensores de dirección lluvia, velocidad del viento, están diseñados para exteriores ya que soportan lluvia, nieve y viento. El resto de los sensores, y el Arduino, necesitan ser protegidos de los factores del tiempo.

² Visitar página web: <https://www.sparkfun.com/products/8942>

Cuesta cerca de \$ 250 y tendrá una estación de código abierto de y moderna donde se tiene el control completo de los componentes por medio de una red Wi-Fi.

Todos los sensores se proveen con cables con terminales RJ11.

Dimensiones: 724mm x 203mm

El kit incluye

- Veleta
- Anemómetro
- Pluviómetro
- Mástil de montaje
- Brazo de montaje del pluviómetro
- Barra de montaje para el anemómetro
- 2 abrazaderas

5.3 Sistema de Monitoreo GrovePi

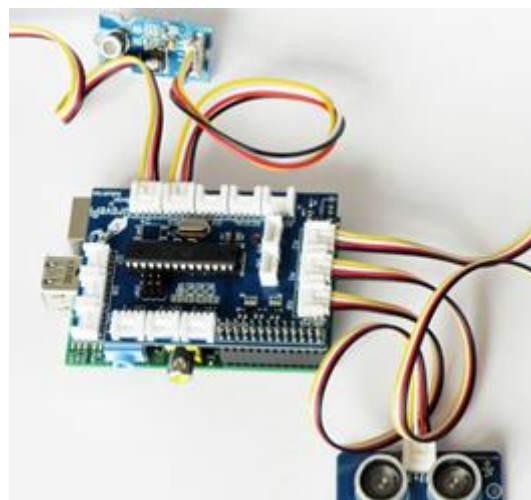


Figura 3: GrovePi

Fuente: <http://www.dexterindustries.com/grovepi/>

GrovePi es una placa electrónica a la cual se puede conectar una gran cantidad de dispositivos ya sea para monitoreo ambiental o para automatización. Posee de 15 interfaces de 4 pines que permiten la conexión de sensores a la Raspberry Pi. Es fácil de usar y modular para conectar hardware con el Raspberry Pi, solo basta con conectarla al Raspberry Pi y se puede empezar a programar. GrovePi es una herramienta fácil de usar que posee una colección de más de

100 módulos plug-and-play de bajo costo que detectan y controlan el mundo físico mediante la conexión de sensores GrovePi a la Raspberry Pi (Dexter Industries, 2016).

El Kit básico esta alrededor de los \$90 y no incluye la Raspberry Pi (Dexter Industries, 2016) y contiene lo siguiente:

- GrovePi³
- Sensor de Sonido
- Sensor de Humedad y Temperatura
- Sensor de Luz
- Relay
- Botón
- Detector de Movimiento
- Leds

³ Este y los demás componentes se pueden encontrar en la página web: <http://www.dexterindustries.com/grovepi/>.

En el siguiente cuadro comparativo se describen las principales características de los sistemas de monitoreo

PROYECTO	PRECIO UNITARIO	ESCALABILIDAD	NUMERO DE SENSORES	IMPLEMENTACIÓN	TECNOLOGIA	CONSUMO DE ENERGIA
AirPi	\$90 (No incluye Raspberry PI)	no es escalable	6 sensores análogos o digitales conectados simultáneamente	La placa Airpi es fácil de implementar que es plug and play hacia un RaspBerry Pi	alámbrica	5v, 900mA, aunque depende de la carga de trabajo de los 4 cores
Weather Meters	\$250(incluye arduino)	no es escalable	3 sensores simultáneamente	Se necesita conocimientos básicos de electrónica	alámbrica	5 v, 900 mA
GrovePi	\$90 (No incluye Raspberry PI)	no es escalable	15 sensores simultáneamente	Se necesita conocimientos básicos de electrónica	alambrica	5v, 900mA, aunque depende de la carga de trabajo de los 4 cores

Tabla 1: Comparativo de Sistemas de monitoreo Ambiental
Fuente: Elaboración Propia (10/01/2016)

5.4 Ventajas

➤ Bajo Costo

Los sistemas que se han considerado son relativamente de bajo costo, no existe mucha diferencia entre ellos y relativamente son muy parecidos en su funcionalidad.

➤ Fáciles de armar y configurar

Los sistemas de monitoreo mencionados son fáciles de armar, basta con conocimientos básicos de electrónica y de informática para ponerlos en funcionamiento en poco tiempo.

➤ Altas tasas de transferencia de datos

El estándar 802.11g permite una velocidad de 54 Mbps y el estándar 802.3 también permite una velocidad de 1000 Mbps lo cual es una altísima tasa de transferencia comparada a los 250 Kbps de ZigBee.

5.5 Desventajas

➤ No son escalables

La comparación de los diferentes sistemas de monitoreo nos da a entender que son poco escalables, es decir si se necesita ampliar el lugar de cobertura es necesario adquirir otra unidad, de igual forma si se necesita aumentar la cantidad de sensores se debería adquirir otra unidad lo cual aumenta los costos.

➤ Son independientes

Al darse la necesidad de querer aumentar el rango de cobertura del área de monitoreo como se mencionó anteriormente se debe adquirir otra unidad de monitoreo lo que conlleva a tener 2 sistemas independientes lo que complica al usuario ya que con estos sistemas no pueden manejar varias unidades desde un mismo sistema, es decir son independientes por cada módulo.

➤ Limitaciones por distancia

Si bien estos sistemas de monitoreo funcionan con el estándar WiFi o Ethernet las distancias son relativamente cortas a las distancias que alcanza el estándar ZigBee, con WIFI sin equipos repetidores se puede alcanzar hasta 500 m con la versión 802.11g y con Ethernet hasta 100m.

6 Configurar los parámetros de los componentes para la red ZigBee y la conexión a Internet.

6.1 Configuración de los módulos Xbee

Para crear una red ZigBee se necesita un nodo coordinador que es el nodo que inicia la propia red, define las políticas de seguridad, maneja las direcciones de los nodos y la auto regeneración de la red. El Gateway Xbee tiene incorporado un módulo ZB Xbee configurado como coordinador en modo AT o modo Transparente, que como se indicaba anteriormente es similar a la comunicación serial y cuando se enciende el dispositivo de puerta de enlace de Xbee ZB se inicializa la red ZigBee, ya que tiene la capacidad de formar redes AD HOC que significa que el coordinador puede formar la red por si solo sin ninguna configuración de una persona.

En la siguiente imagen se indica que el coordinador determinó la PAN ID con la dirección 0x7119 y si se quiere unir cualquier nodo pues tendrá que tener el mismo PAN ID. El XBee Gateway también está configurado para utilizar cualquiera de los 16 canales, por lo que el coordinador puede realizar una búsqueda de canales y escoger la mejor opción para la red.



Figura 4: Configuración Xbee
Fuente: Elaboración Propia (10/01/2016)

6.2 Configuración del módulo Coordinador

El coordinador tiene la dirección 00:13:a2:00:40:c9:97:c2. En este caso es una dirección compuesta por dos direcciones la DH o Destination High que son los 8 bits más y los siguientes 8 bits menos significativos para conformar la dirección Destination Low. Los 16 bits en conjunto forman

una dirección única, es decir no existe otro Xbee con esta dirección y para la identificación en la red PAN Xbee usa los 8 bits menos significativos es decir 40:c9:97:c2., o la DL. Estos datos también se observan en la parte inferior de Gateway Xbee y no son direcciones configurables, sino que vienen grabadas de fábrica.



Figura 5: Dirección DH y DL del Gateway Xbee
Fuente: Elaboración Propia (10/01/2016)

En el menú *Network Settings* se observa la configuración básica del módulo coordinador como indica la siguiente pantalla:

Figura 6: Configuración Básica del Módulo Xbee Coordinador
Fuente: Elaboración Propia (10/01/2016)

El campo Extended PAN ID está configurado en 0 ya que no es necesario usarlo por que basta con los 4 bits del campo Pan ID para identificar nuestra red. En este caso el coordinador eligió la dirección 0x7119 de manera aleatoria pero es completamente configurable según lo necesite el usuario.

Por otro lado, el campo Extended PAN ID es útil cuando en el perímetro existen varias redes Xbee y los 4 bits del campo PAN ID no sean suficientes para seguir aumentando redes. La siguiente pantalla nos indica nuestro campo PAN ID y la dirección DH y DL del Gateway

```
XBee Device on the Gateway
Network PAN ID: 0x7119 - 0x67bf8d8b78e1347e
Network Channel: 0x12 (2440 MHz)
Gateway XBee Address: 00:13:a2:00:40:c0:97:c2!
Gateway XBee Firmware: 0x4043
```

Figura 7: Campo PAN ID
Fuente: Elaboración Propia (10/01/2016)

ZigBee divide la banda 2.4 Ghz en 16 canales, los canales que no están activados el coordinador no los escanea. En este caso los canales que no están seleccionados son aquellos que la versión del módulo Xbee no soporta, éstos están disponibles para las nuevas versiones de módulos Xbee.

Scan Channels (SC): bitfield (0x1-0xffff)

☐ 0b ☒ 0c ☒ 0d ☒ 0e ☒ 0f ☒ 10 ☒ 11 ☒ 12
☒ 13 ☒ 14 ☒ 15 ☒ 16 ☒ 17 ☐ 18 ☐ 19 ☐ 1a

Figura 8: Calanes disponibles para selección
Fuente: Elaboración Propia (10/01/2016)

En este caso se los puede definir activando los canales que se van a escanear según el módulo que disponemos o directamente mediante el campo *Scan Chanel* con el valor en hexadecimal según la siguiente matriz:

Channel/Mask	Notes	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11
0x1FFE	Best for mixed old/new Xbee networks, very safe	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0

Figura 9: Distribución de Canales
Fuente: Digi¹

Es decir 0001111111111110 = 1FFE Hex, Como ejemplo se podría decir que si quisiera activar solo el canal 26 se digitaría el número 1000000000000000 sería 0x8000 la máscara en hex.

6.3 Configuración del módulo Router

El módulo Router tiene la dirección 00:13:a2:00:40:bf:d9:64, como se explicó anteriormente, es una dirección compuesta por las direcciones DH y DL. Ya que la red solo va a estar identificada por el campo PAN ID y el campo Extended PAN ID, va a estar en 0 como muestra la siguiente pantalla:

The screenshot shows the 'Network Settings' configuration page. It includes the following fields and options:

- Extended PAN ID (ID):** A text box containing '0x0000000000000000' with a label '8 hex bytes'. Below it, text reads: 'Enter 0 to join a network with any extended PAN ID. Note: Changing the PAN ID may make this device inaccessible.'
- Scan Channels (SC):** A text box containing '0x7fff' with a label 'bitfield (0x1-0xffff)'. Below it, a grid of checkboxes for channels 0b through 1a is shown. Channels 0b through 19 are checked, while 1a is unchecked.
- Scan Duration (SD):** A text box containing '3' with a label '(0-7)'.
- ZigBee Stack Profile (ZS):** A dropdown menu showing 'Network Specific (0)'.
- Node Join Time (NJ):** A text box containing '255' with a label 'seconds (0-255, 255=always)'.
- Network Watchdog Timeout (NW):** A text box containing '0' with a label 'minutes (0-25855, 0=disabled)'.
- Channel Verification (JV):** A checkbox labeled 'Enabled' which is checked.
- Join Notification (JN):** A checkbox labeled 'Enabled' which is unchecked.
- Device Options (DO):** A text box containing '0x1' with a label 'bitfield (0-0xff)'. Below it, two checkboxes are shown: 'Enable Temperature Compensation Updates' (checked) and 'Enable Best Response Joining' (unchecked).

Figura 10: Configuración Básica del Módulo Xbee Router
Fuente: Elaboración Propia (10/01/2016)

Por lo tanto, en el campo PAN ID debe ir la misma dirección de red que selecciona el coordinador.

La división de los canales en la frecuencia 2.4 Ghz en el caso del router se puede observar que, a diferencia del módulo coordinador, este puede trabajar en más canales, en 15 de los 16 canales totales, por lo cual escanea los 15 permitidos, esto se debe a que es un módulo Xbee Serie2 PRO más actual que el coordinador.



Figura 11: Selección de Canales
Fuente: Elaboración Propia (03/03/2016)

En la imagen se observa que al usar la dirección 00:00:00:00:00:00:00:00 de la dirección extendida de DH y DL son 0. En este punto de la configuración se puede inferir cómo se va a comportar nuestro módulo router, dependiendo de la topología de la red.

Para ello, existen tres opciones:

A. Configurar la Destination Address en 00:00:00:00:00:00:00:00

Con esta configuración todos los datos del nodo los envía al coordinador, en el caso de estudio es importante, ya que el nodo coordinador es el que sirve de enlace con el Internet.

B. Configurar la Destination Address en 00:00:00:00:00:00:ff:ff

Con esta configuración los datos del nodo se envían a cada nodo de la red, por ejemplo si en un parque se quisiera encender todas las luminarias, se enviaría la orden a todos los módulos Xbee que controlan las luminarias, es decir hace un broadcast de datos.

C. Configurar la Destination Address con la información de un nodo

Cuando se necesite enviar información a cualquier nodo ya sea coordinador, router o end device de la red se configura el campo *Destination Address* con la dirección que forma DH y DL, en este caso tendría el campo *Destination Address* la dirección del coordinador 00:13:a2:00:40:c0:97:c2.

Para el caso de estudio se deja configurada en 00:00:00:00:00:00:00:00 es decir la información se envía al coordinador como muestra la pantalla.

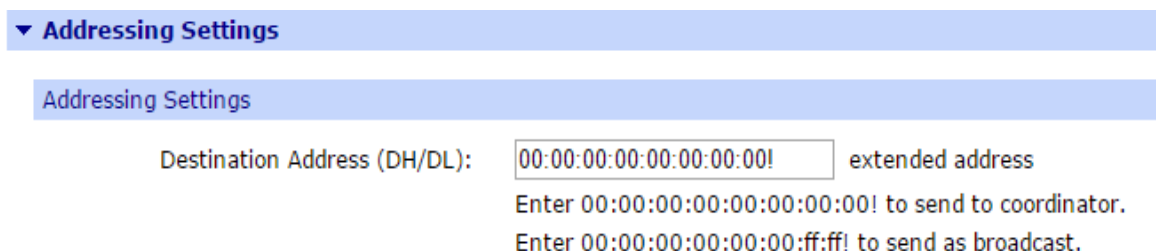


Figura 12: Configuración DH/DL extendida
Fuente: Elaboración Propia (03/03/2016)

6.4 Configuración de módulos Xbee con el programa XCTU

Existen diferentes métodos de configuración de los módulos Xbee, básicamente todos los programas que conforman la comunicación por puerto serial. Por ejemplo Hyperterminal. Para este ejemplo se va a usar el programa XCTU, que es en modo gráfico y muy fácil de entender para el usuario. Este se puede descargar desde el siguiente link: <http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/xctu>.

X-CTU es una aplicación para MAC y Windows que provee una interfaz gráfica para la configuración e interacción con los módulos Xbee, para esto se necesita un programador de módulos Xbee con puerto USB, en este caso el que se muestra en la siguiente figura.



Figura 13: Programador con Módulo Xbee S2 Pro
Fuente: Elaboración Propia (03/03/2016)

Este es un programador con fines didácticos, el cual incluye una barra de leds, acelerómetro, un potenciómetro, entre otros elementos.

La siguiente pantalla es la que da inicio a la aplicación, en donde tan solo se debe hacer clic en el signo + para hacer la conexión con el programador USB.



Figura 14: Aplicativo XCTU
Fuente: Elaboración Propia (10/01/2016)

En la siguiente pantalla se muestran los parámetros de la conexión para la comunicación serial y para la configuración de los módulos Xbee.

- Baud Rate: 9800
- Data Bits: 8
- Bit de parada: 1

Select the Serial/USB port:

	COM4	USB Serial Port

Refresh ports

Baud Rate: 9600

Data Bits: 8

Parity: None

Stop Bits: 1

Flow Control: None

Figura 15: Parámetros para la comunicación serial
Fuente: Elaboración Propia (03/03/2016)

Una vez hecha la conexión se puede observar el módulo Xbee listo para configurarlo. En este paso se configurará el módulo Xbee router en modo transparente. Como se explicó anteriormente, este es el modo más fácil de configurar un módulo Xbee.

Los parámetros que hay que tener en cuenta para configurar el Xbee en modo transparente son que el coordinador también este configurado en modo transparente o modo AT. Para llegar al modo transparente hay que tener en cuenta lo siguiente:

1. **DH:** La Dirección de destino alta o DH es el número estándar del módulo xbee, y se encuentra al revés del Xbee por lo general es el 13A200.
2. **DL:** La dirección de destino baja es único del módulo Xbee, este lo encontramos en la parte baja de la dirección DH para este módulo es el 40BFD964.



Figura 16: Dirección DH y DL impresa en el módulo Xbee
Fuente: Elaboración Propia (03/03/2016)

En la siguiente pantalla nos muestra la conexión con el módulo Xbee router

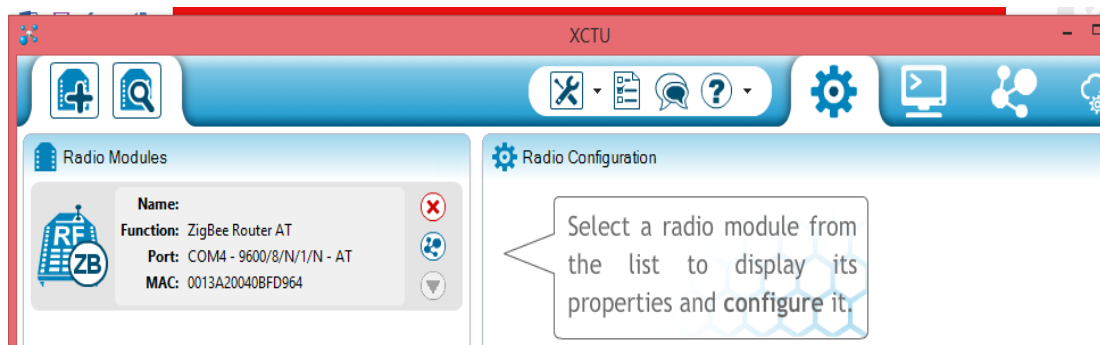


Figura 17: Conexión del módulo Xbee en XCTU
Fuente: Elaboración Propia (03/03/2016)

Con XCTU se muestra la configuración básica para una conexión punto a punto con el módulo Xbee interno del GateWay Xbee. Como se indicaba anteriormente, cuando se ingresaba al modo de configuración por browser del GateWay Xbee, el campo Extended PAN ID debe configurarse en 0 y en el campo SC Scan Channels en 7FFF, ya que con ello se indica que se van a scanear los primeros 15 canales, anteriormente explicados (Digi, 2015).

En la siguiente figura se muestran campos básicos a configurar en una red punto a punto.

ID PAN ID	0
SC Scan Channels	7FFF Bitfield
SD Scan Duration	3 exponent
ZS ZigBee Stack Profile	0
NJ Node Join Time	FF x 1 sec
NW Network Watchdog Timeout	0 x 1 minute
JV Channel Verification	Enabled [1]
JN Join Notification	Disabled [0]
OP Operating PAN ID	0180D08B78E1347E
OI Operating 16-bit PAN ID	7119
CH Operating Channel	12
NC Number of Remaining Children	C

Figura 18 : Configuración para la conexión punto a punto básica
Fuente: Elaboración Propia (03/03/2016)

El direccionamiento se lo configura en los DH y DL en 0 de la misma forma que en la configuración realizada anteriormente por Browser, y en consecuencia del módulo Xbee al coordinador automáticamente sin necesidad de colocar su dirección DH y DL. Esto es útil cuando no tenemos acceso al coordinador para ver la dirección físicamente (ARDUINO, 2015).

Los campos SH o *Serie High* y SL o *Serie Low* determinan de qué módulo se va a leer la información es decir son la dirección DH y DL del módulo a transmitir la información (ARDUINO, 2015).

Las siguientes configuraciones mostradas nos aseguran la comunicación con nuestro coordinador y se pueden ver reflejadas en la pantalla de configuración del Gateway Xbee.

Figura 19: Configuración básica para la conexión con el coordinador directamente
Fuente: Elaboración Propia (20/03/2016)

▼ Addressing
Change addressing settings

① SH Serial Number High	13A200	⊙	⊗
① SL Serial Number Low	40BFD964	⊙	⊗
① MY 16-bit Network Address	FFFE	⊙	⊗
① DH Destination Address High	0013A200	⊙	⊗
① DL Destination Address Low	40C097C2	⊙	⊗
① NI Node Identifier		⊙	⊗
① NH Maximum Hops	1E	⊙	⊗
① BH Broadcast Radius	0	⊙	⊗

Figura 20: Configuración básica para la conexión punto a punto con cualquier otro nodo
Fuente: Elaboración Propia (20/03//2016)

Con la Aplicación XCTU también se puede ver la distribución de pines del nodo Router, así se puede cambiar su modo de funcionamiento, ya sea analógica o digital, dependiendo del pin y del tipo de sensores o dispositivo que se tenga conectado. Si es un sensor con envío de información serial se usaría una entrada analógica, por el contrario si se tiene conectados switchs se usaría una entrada digital (Digi, 2015).

▼ I/O Settings
Modify DIO and ADC options

① D0 AD0/DIO0 Configuration	Commissioning Button [1]	⊙	⊗
① D1 AD1/DIO1 Configuration	ADC [2]	⊙	⊗
① D2 AD2/DIO2 Configuration	ADC [2]	⊙	⊗
① D3 AD3/DIO3 Configuration	ADC [2]	⊙	⊗
① D4 DIO4 Configuration	Digital Input [3]	⊙	⊗
① D5 DIO5/Assoc Configuration	Associated indicator [1]	⊙	⊗
① P0 DIO10/PWM0 Configuration	Digital Out, Low [4]	⊙	⊗
① P1 DIO11 Configuration	Digital Out, High [5]	⊙	⊗
① P2 DIO12 Configuration	Digital Out, Low [4]	⊙	⊗
① PR Pull-up Resistor Enable	3DFF	⊙	⊗
① LT Associate LED Blink Time	0 x10 ms	⊙	⊗
① RP RSSI PWM Timer	28 x 100 ms	⊙	⊗
① DO Device Options	1 Bitfield	⊙	⊗

Figura 21: Configuración pines analógicos y digitales
Fuente: Elaboración Propia (20/03/2016)

7 Elaboración del Prototipo

El monitoreo del ambiente consiste en utilizar métodos de medición para describir el estado del ambiente y para realizar un pronóstico para la toma decisiones, el monitoreo ambiental tiene un

campo muy amplio de aplicaciones, así puede monitorear el ambiente de una habitación hasta el monitoreo de una región.

Dependiendo del tipo de monitoreo a realizarse se debe elegir el tipo de sensor ya que los sensores pueden ser de 2 tipos físicos y químicos, los físicos miden variables como el calor, la luminosidad, la presión, etc. Los sensores químicos miden la presencia de gases, la acidez del suelo, la presencia de algún componente en líquidos.

En este caso de estudio se va a utilizar 2 sensores físicos y analógicos que midan la luminosidad ambiental y la temperatura, el sensor de luz o fotocelda que se va a usar es el GL5528 que es suficiente para medir la luminosidad de un mini invernadero.



Figura 22: Fotocelda
Fuente: <https://www.sparkfun.com>

El sensor de temperatura a usar es el TMP36 que es un sensor analógico sencillo de implementar y suficiente para registrar la temperatura de una habitación siendo en este caso adecuado para el mini invernadero.



Figura 23: Sensor de Temperatura
Fuente: <https://www.sparkfun.com/>

Para la configuración de la red Zigbee se va a constituir de 2 elementos un módulo Xbee S2 B que transmite a una distancia de 200 metros a 250 Kbps, capaz de implementar 6 sensores por módulo Xbee, este módulo es suficiente para la implementación de los dos sensores de temperatura y luminosidad. Es capaz de ser configurado también su tiempo de actividad para prolongar su tiempo de vida activa y ahorrar energía.



Figura 24: Modulo Xbee Pro S2

Fuente: <https://www.digi.com/>

Usaremos un Xbee Gateway que es el enlace entre la red ZigBee y el Internet, este dispositivo tiene internamente un módulo Xbee configurado como coordinador, si es necesario implementar más módulos Xbee se los debe configurar con los datos de este dispositivo para enviar la información. Trabaja de dos interfaces de red una Ethernet y la otra WiFi haciendo fácil la implementación y configuración para el usuario.



Figura 25: Gateway

Fuente: <https://www.digi.com/>

Los sensores son pequeños dispositivos que según sus características captan variables ya sean físicas o químicas y las transforman en corriente eléctrica que, según sus variaciones eléctricas dan una idea de la magnitud de la variable ya sea temperatura, humedad, movimiento, aceleración, etc., y variables químicas como son el PH, niveles de CO₂, humo y detección de gases.

Para este caso de estudio vamos a utilizar 2 sensores analógicos para registrar los niveles de luz y un sensor de temperatura cuya configuración se la describe a continuación:

7.1 Implementación del sensor de luz

Para ensamblar el circuito vamos a usar las siguientes partes:

- Un sensor de luz o Foto celda⁴
- 3 resistencias de 10 k ohm⁵
- Cables⁶

Para la configuración del módulo se va a usar la aplicación XCTU para conectarnos a el nodo router que es el que va a enviar la información de la foto celda. Se necesita configurar la entrada y salida de información del módulo Xbee, la foto celda envía la información con una señal análoga, por lo que el pin D1 se configura para que reciba la señal análoga, como indica la imagen (Digi, 2015).

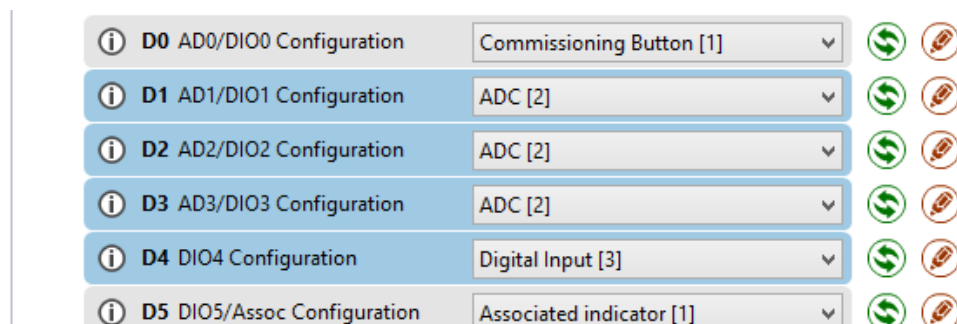


Figura 26: Configuración de Pines
Fuente: Elaboración Propia (20/03/2016)

El diseño del circuito se realizó con el programa “Fritzing” que es un aplicativo para el desarrollo de circuitos. El circuito tiene el siguiente esquema:

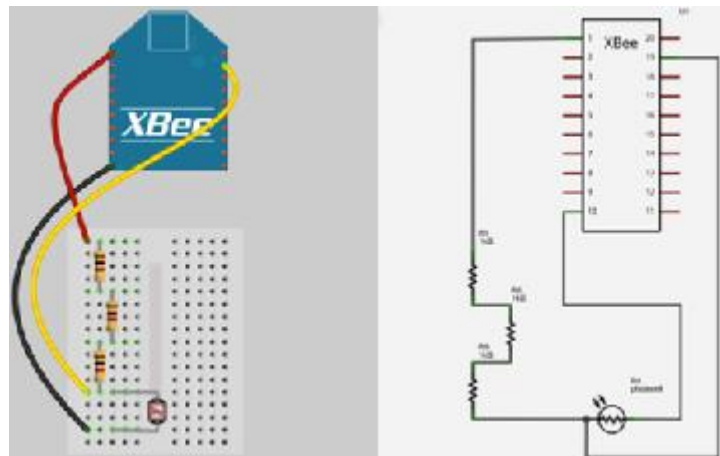


Figura 27: Esquema del circuito de la fotocelda
Fuente: Elaboración Propia (20/03/2016)

⁴ Revisar página web: <http://examples.digi.com/xbee-gateway/light-sensor-example-xbee-zigbee-cloud-kit/>

⁵ Revisar página web: <http://examples.digi.com/xbee-gateway/light-sensor-example-xbee-zigbee-cloud-kit/>

⁶ Revisar página web: <http://examples.digi.com/xbee-gateway/light-sensor-example-xbee-zigbee-cloud-kit/>

En la siguiente figura nos muestra el circuito armado en nuestra placa.

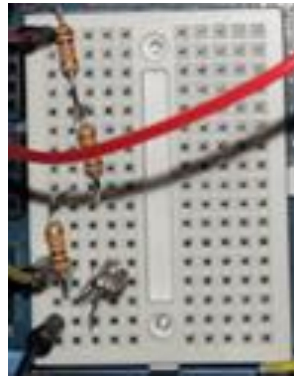


Figura 28: Circuito de la Fococelda
Fuente: Elaboración Propia (20/03/2016)

Configuración del Widget para el sensor de luz

En la siguiente imagen se indican los parámetros necesarios para crear un Widget, el cual mostrará la información de la fotocelda. Se tiene que crear el Widget con los siguientes datos:

- Se le da un título al Widget
- Se ingresa el ID del Módulo Xbee Router
- Seleccionar ADC1 como el flujo de entrada análoga
- Se selecciona la unidad en este caso porcentaje
- Se establece un valor mínimo de 0 y un máximo de 100 para ver el rango de luz porcentaje mostrado.
- Se ingresa la formula " $100 - (\text{value} / 1000 * 100)$ " para transformar la entrada de mili voltios a la unidad a porcentaje de 0 a 100. La fórmula tiene el valor de entrada con la variable "value" descrita el código JavaScript que se encuentra en el Anexo 1. Dado que el valor resultante enviado por la fotocelda no es la intensidad de luminosidad sino es el valor de oscuridad, se hace una resta de 100 para obtener el porcentaje de luminosidad.

Widget Settings

Widget Type	Gauge Widget
Label*	<input type="text" value="Light"/>
Device*	<input type="text" value="00:40:9D:5E:30:38"/>
Input Stream*	<input type="text" value="ADC1"/> <button>Check Device Configuration</button>
Input Transform	<input type="text" value="100 - (value / 2500 * 100)"/>
Units	<input type="text" value="percent"/>
Low value	<input type="text" value="0"/>
High value	<input type="text" value="100"/>

* Required field

Figura 29: Parametrización del Widget
Fuente: Elaboración Propia (20/03/2016)

Se salva los cambios y se puede observar en la web el Widget, con la información enviada desde la fotocelda, a través del módulo Xbee por una conexión serial hacia el Gateway Xbee y desde el Gateway hacia el internet en la siguiente pantalla:



Figura 30: Widget para la medición de luminosidad
Fuente: Elaboración Propia (20/03/2016)

7.2 Implementación del sensor de temperatura

La Medición de la temperatura es una forma popular para empezar con sensor analógico. En este ejemplo se utiliza el sensor lineal de baja tensión TMP36 que es muy fácil de configurar.

El sensor genera una tensión de salida que es directamente proporcional a la temperatura Celsius. Cuanto más caliente es, mayor es el voltaje que se pasa de analógico a digital de la Xbee (ADC) (Digi, 2015).

Para la creación del circuito se necesitará tres componentes:

- Un sensor TMP36⁷
- 4 resistencias de 1 k ohm
- Cables para la conexión

También se necesita configurar una de 3 entradas analógico digital en el Xbee, donde va a estar conectado el sensor de temperatura, esta entrada se la configura como entrada de señal analógica como muestra la siguiente figura (Digi, 2015).

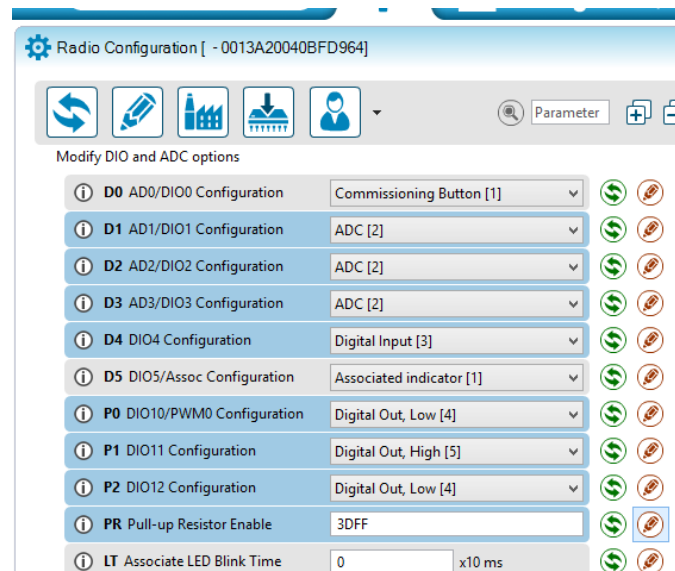


Figura 31: Configuración de Pines Analógicos
Fuente: Elaboración Propia (20/03/2016)

En la misma página, también se configura la frecuencia de muestreo que se establece en 5000 ms y que tendrá una muestra de cada cinco segundos.

⁷ Este y los demás componentes se pueden encontrar en la página web: <http://examples.digi.com/xbee-gateway/temperature-example-xbee-cloud-kit/>

Esquema del Circuito

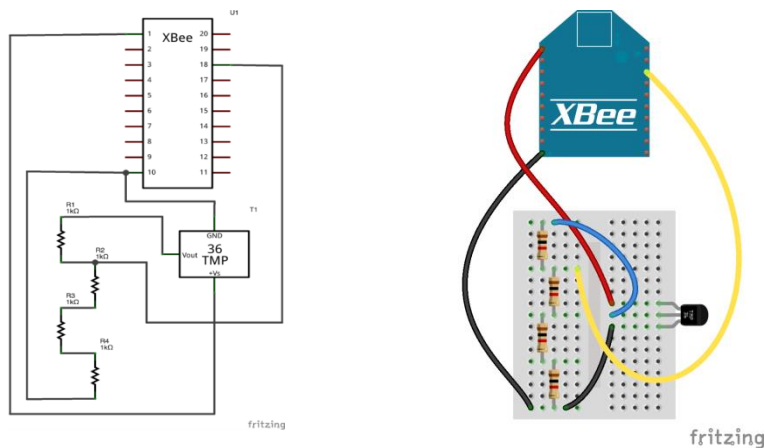


Figura 32: Diseño del circuito para el sensor de temperatura
Fuente: Elaboración Propia (10/01/2016)

Para la creación del circuito se necesita lo siguiente:

- Se conecta el sensor de temperatura en las tres filas separadas en el protoboard, como se muestra en la Figura 33. Con la parte plana frente a los cables, los pines se numeran del 1 al 3 de derecha a izquierda (Digi, 2015).
- Se conecta un cable del pin de energía en la misma fila que el pin 1 del TMP36 a 3.3 voltios de potencia (Digi, 2015).
- Se conecta un cable de puente negro para que el pin 3 del TMP36 está conectado a GND (tierra) (Digi, 2015).
- Se conecta las cuatro resistencias en filas separadas del tablero, como se muestra en la figura 42, así se formara una cadena con cada resistencia que se conecta a uno de los extremos de cada resistencia a ambos lados (Digi, 2015).

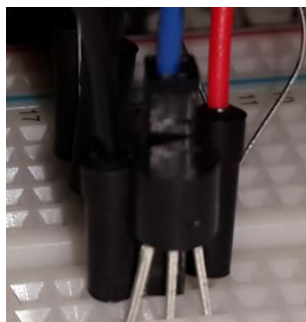


Figura 33: Conexión del sensor de Temperatura
Fuente: Elaboración Propia (10/01/2016)

- Se conecta el cable del pin 2 del sensor TMP36 está conectado al extremo de la primera resistencia (Digi, 2015).
- Se conecta un cable a la fila en la que las primera y segunda resistencias se encuentran. Conecte el otro extremo de este cable al pin AD2 del Xbee (Digi, 2015).
- Se conecta extremo de la cuarta resistencia a GND (Digi, 2015).

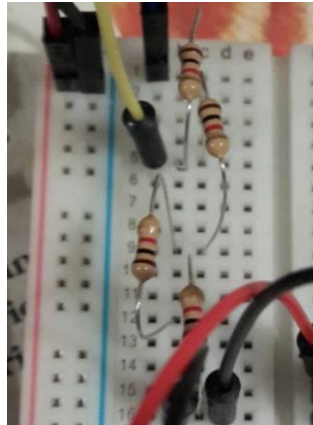


Figura 34: Conexión de las resistencias
Fuente: Elaboración Propia (10/01/2016)

Para crear el Widget de temperatura se necesitan los siguientes parámetros:

- Seleccione ADC2 como el flujo de entrada serial.
- Ingresar " value / 0,75 / 10-50 " para transformar la entrada de mili voltios a grados Celsius, donde value es una variable la cual se explica en el código de programación que se encuentra en el Anexo 1.
- Se establece un valor mínimo de 0 y un máximo de 50 para ver un rango de temperatura Celsius.
- Si desea utilizar Fahrenheit , el cálculo es "(valor / 0,75 / 10-50) * 1.8 + 32 "

Widget Settings

✕ Remove Widget

Widget Type	Gauge Widget
Label*	Temperature
Device*	00:40:9D:5E:30:38
Input Stream*	ADC2 <input type="button" value="Check Device Configuration"/>
Input Transform	value / 0.75 / 10 - 50
Units	°C
Low value	0
High value	150

Figura 35: Creación del Widget de temperatura
Fuente: Elaboración Propia (20/03/2016)

Y nuestro DashBoard quedaría de la siguiente manera:

XBee ZigBee Cloud Kit



Figura 36: Dashboard para el monitoreo del invernadero
Fuente: Elaboración Propia (20/03/2016)

5.2.3 Implementación módulo chat

La implementación de un sistema de chat nos ayuda a enviar y recibir información remota en mensajes de texto, de este modo se puede dar cualquier instrucción desde un browser en cualquier parte del mundo con acceso a internet. Con el chat se demuestra que no solo se puede manejar o enviar información de sensores a través de los módulos Xbee sino cualquier información que se pueda enviar desde una conexión serial.

Para este proyecto se usó lo siguiente:

- Un equipo Raspberry Pi (o puede ser cualquier computador) para enviar y recibir los mensajes.
- Tener instalado en el Raspberry Pi una consola como el HyperTerminal o putty o cualquier programa que permita realizar conexiones seriales.
- Acceso a un Browser con internet, para enviar y recibir mensajes en cualquier parte del mundo.

Para la creación del Widget para el envío de texto se necesita lo siguiente:

En la siguiente imagen nos indica los parámetros necesarios para crear un Widget que nos mostrará la información de la fotocelda. Se tiene que crear el Widget con los siguientes datos:

- Se da un título al Widget en la etiqueta Label que en este caso se puso “MENSAJES”
- En la Etiqueta Gateway se ingresa el ID del Módulo Xbee Router
- En la etiqueta Xbee Module se ingresa el ID del Módulo Xbee que va enviar o recibir los mensajes de Texto.

Figura 37: Creación del Widget del módulo de Mensajes
Fuente: Elaboración Propia (20/03/2016)

Una vez creado el widget de mensajería nuestro dashboard queda de la siguiente manera:

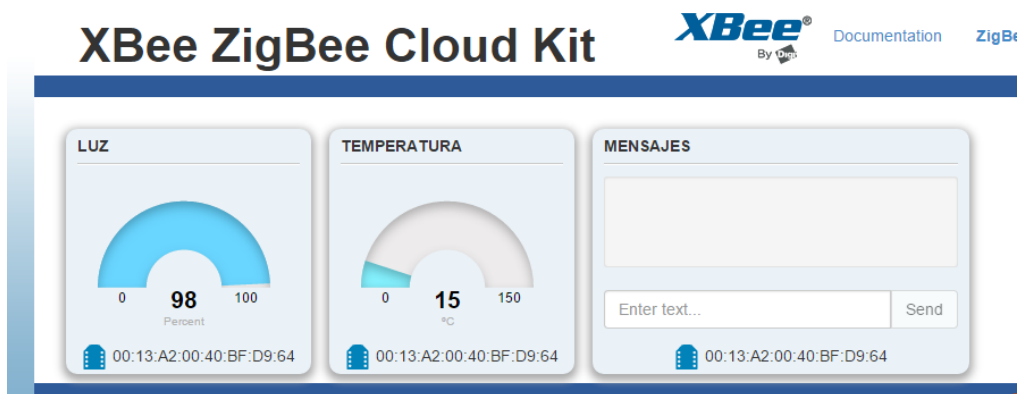


Figura 38: Dashboard para el monitoreo Ambiental
Fuente: Elaboración Propia (20/03/2016)

7.3 Implementación, funcionamiento del equipo de monitoreo dentro del invernadero.

En este capítulo se implementará el sistema de monitoreo ambiental en un mini invernadero para demostrar su funcionalidad. Se monitorearán dos factores que son la luminosidad y la temperatura, también se montará un equipo Raspberry Pi, con una webcam para tener acceso visual al invernadero y para tener una terminal de consola para el envío de mensajes de texto, a través de un browser y un terminal serial en Putty. Con esto, se pretende demostrar la versatilidad del sistema ya que servirá solo para el propósito original que es el monitoreo ambiental, sino que si se necesita otras funcionalidades con los módulos de Xbee pueden implementarse.

Para este caso de estudio se ha elegido probar el sistema de monitoreo ambiental para un mini invernadero del cultivo de la especie de planta *Dionea Muscipula* o más conocida como la Venus Atrapamoscas. Esta es una planta originaria de Carolina del Norte en Estados Unidos, donde las condiciones para su desarrollo son muy diferentes a las condiciones climáticas en Ecuador ya que se tiene diferente fotoperiodo (Horas de Luz y Horas de Oscuridad), diferentes estaciones y diferentes temperaturas, ya que en su lugar de origen la temperatura son muy marcadas y muy variadas durante el año. En cambio en Ecuador son casi constantes en todo el año.

A pesar de las condiciones ambientales autóctonas de las venus atrapamoscas, se han visto pequeños cultivadores que actualmente las están comercializando en nuestro país siendo muy importante el monitoreo ambiental de esta especie para un buen desarrollo. En todo el año las venus atrapamoscas necesitan cumplir 3 ciclos que se explicaran a continuación.

➤ **Ciclo Vegetativo o de crecimiento**

En esta fase la venus atrapamoscas comienza a generar follaje, ganar tamaño y aumentar el número de trampas para poder alimentarse, esto ocurre en verano cuando el día es más largo que la noche, en su lugar de origen se mantiene con 16 horas de luz y 8 horas de oscuridad.

➤ **Ciclo de Floración**

El ciclo de floración empieza en primavera cuando los niveles de luz se van reduciendo hasta llegar a un fotoperiodo de 12 horas de luz y 12 horas de oscuridad este es el foto periodo en la zona ecuatorial todo el año.

En este momento la planta empieza a florar por tal motivo consume todas las energías que se han guardado en el período vegetativo la planta produce flores y están listas para la reproducción y generación de semillas.

➤ **Ciclo de hibernación**

Una vez que la planta ha gastado todas sus energías produciendo sus flores y semillas la planta queda débil, justo en esta época es cuando la temperatura ambiental baja menos de 10 grados centígrados y está preparada para descansar, baja su metabolismo y espera hasta el próximo verano para volver al ciclo vegetativo nuevamente.

Como se puede dar cuenta en todos los ciclos es fundamental controlar las horas de luz y la temperatura para así asegurar la correcta producción de este tipo de plantas.

7.4 Armado del sistema de gestión del Mini invernadero

Los elementos que utilizaremos son:

- Módulo Xbee Router
- GateWay (Xbee Coordinador)
- Circuito instalado Fotocelda
- Circuito instalado sensor de temperatura
- RaspBerry Pi (Para el monitoreo por webcam y envío de información serial)
- Webcam

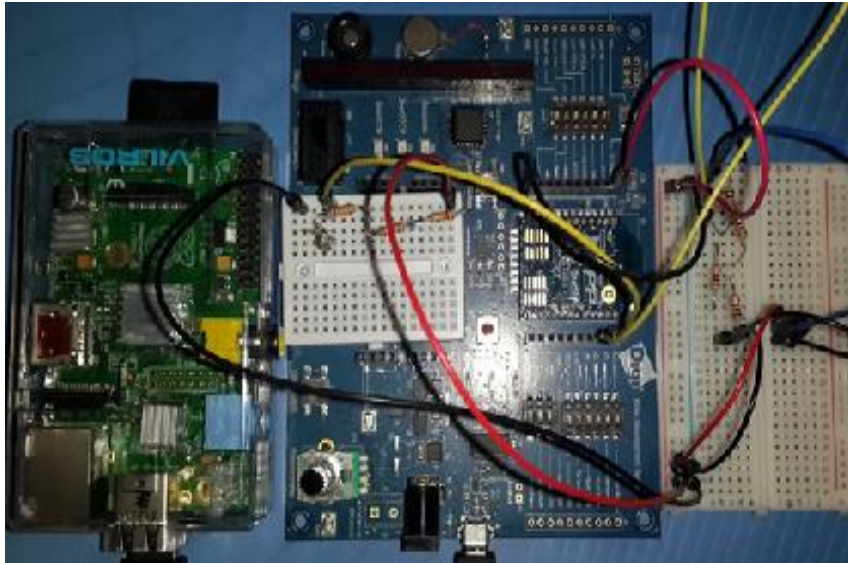


Figura 39: Equipos para la comunicación y monitoreo de invernadero
Fuente: Elaboración Propia (20/03/2016)

- Con todos los elementos listos se procede a colocar los circuitos armados junto con nuestro Xbee Router dentro del invernadero.
- Se conecta el módulo Xbee al equipo Raspberry Pi ya que servirá para energizar los circuitos y el módulo Xbee Router, como punto de acceso remoto para la visualización del invernadero y los mensajes de texto de la comunicación serial.
- Se conecta la Webcam al Raspberry Pi donde previamente tiene que tener instalado el aplicativo que permite visualizar la imagen de la cámara, el aplicativo se llama *luvview* y se lo descarta desde un terminal con el comando **sudo apt-get install luvview**
- Se instala el aplicativo para la conexión remota al Raspberry Pi, en este caso se usa *XRDP* y se lo descarta desde un terminal con el comando **sudo apt-get install xrdp**

Realizada la conexión nos muestra la interfaz serial de la siguiente forma:

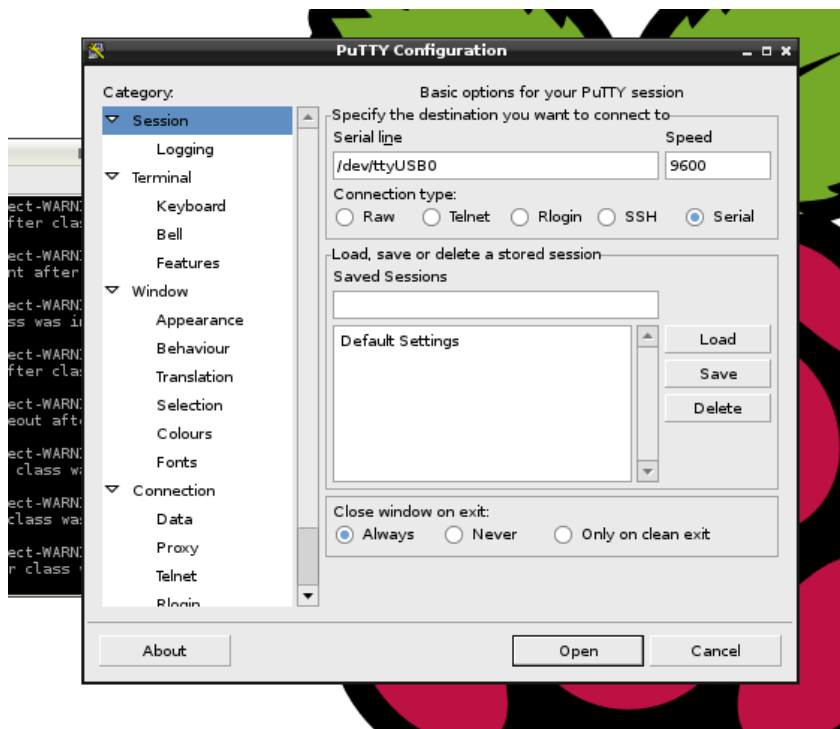


Figura 40: Parámetros para la conexión serial hacia el módulo Xbee
Fuente: Elaboración Propia (20/03/2016)

7.5 Puesta en marcha del Sistema de monitoreo ambiental

La implementación de este sistema consiste en monitorear un mini invernadero de plantas, exactamente la especie Venus Atrapamoscas. Por lo general éstas se cultivan en invernaderos cuando se encuentran fuera de su hábitat, ya que se debe cuidar con detalle ciertos parámetros para simular su hábitat y alargar su tiempo de vida.

Las condiciones óptimas para su desarrollo es tener al menos 15h de luz al día y una temperatura de hasta 10 grados centígrados en la noche y 25 grados centígrados en el día. Para el pleno desarrollo la planta debe cumplir 3 ciclos que depende netamente del medio ambiente.

La temperatura es parte fundamental para el correcto funcionamiento de las plantas ya que si por varios días la temperatura desciende menos de 10 grados. Para monitorear la cantidad de luz y temperatura se encuentran los sensores dentro del invernadero junto con una webcam que servirá para visualizar las plantas. La webcam se conecta directamente al Raspberry Pi al cual se podrá ingresar vía remota para observar el invernadero y los mensajes de la conexión serial.

La siguiente imagen muestra los sensores y el módulo Xbee dentro del invernadero

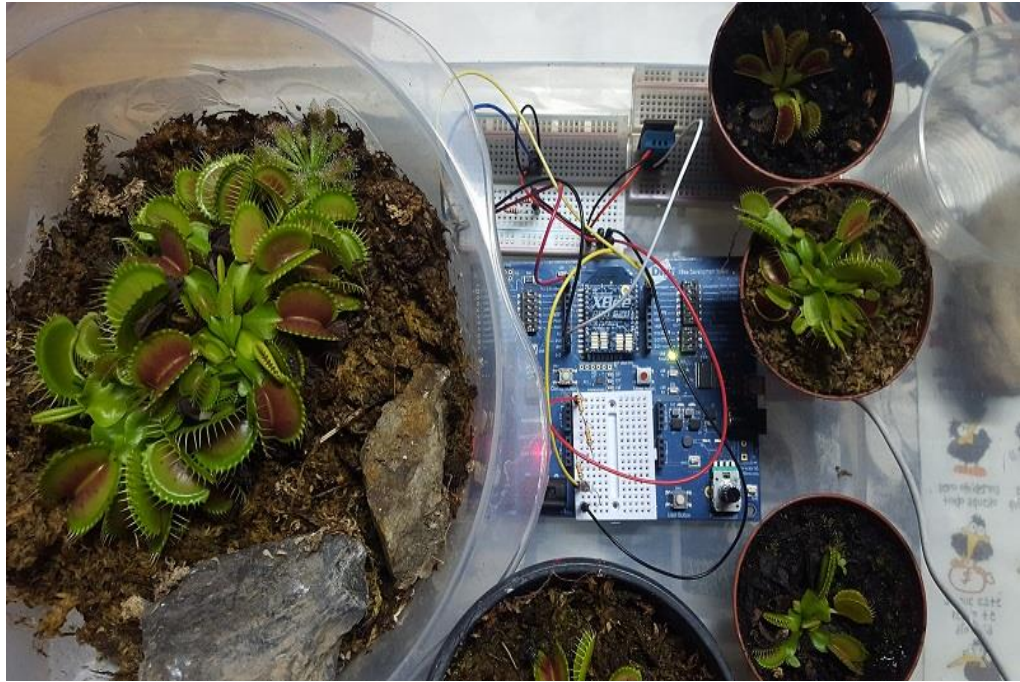


Figura 41: Sensores y Xbee Router dentro del invernadero
Fuente: Elaboración Propia (20/03/2016)

En la siguiente figura se muestra el funcionamiento integral del ejemplo

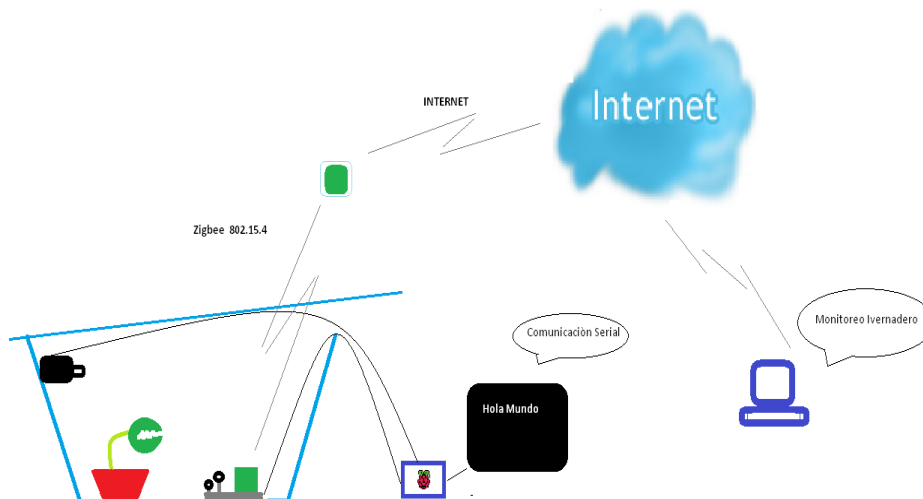


Figura 42: Esquema general del monitoreo del invernadero por ZigBee e Internet
Fuente: Elaboración Propia (11/04/2016)

La siguiente imagen corresponde a la conexión remota al Raspberry Pi donde se instaló el visor *LUVView* para proyectar las imágenes que captura la webcam. Se observa la ventana de la interfaz serial de Putty que se conecta con el Módulo Xbee Router con los siguientes parámetros:

- Serial Line: `/dev/ttyUSB0`

- Speed: 9600
- Conexión: Serial

Una vez hecha las dos conexiones en la siguiente figura se visualizará el invernadero a través de la webcam y también la interfaz serial donde se dejaran los mensajes de texto que se pueden enviar desde el entorno web de monitoreo.

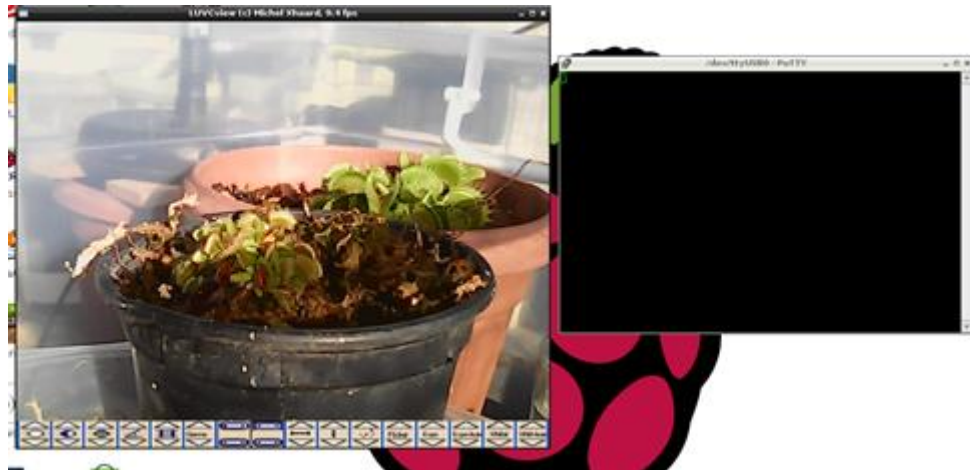


Figura 43: Esquema general del monitoreo del invernadero por ZigBee e Internet
Fuente: Elaboración Propia (10/01/2016)

Posteriormente, se realiza el monitoreo del invernadero con los sensores de luz y de temperatura. Para esto ingresamos al link: <https://xbeegateway.herokuapp.com/#/login>.

Para ingresar al Dashboard de monitoreo, se utilizan los siguientes datos:

- Usuario: zigbeeinternet
- Contraseña: MI123456!

Una vez ingresados los datos, se puede monitorear la intensidad lumínica y la temperatura del invernadero, como se presenta en la siguiente pantalla:

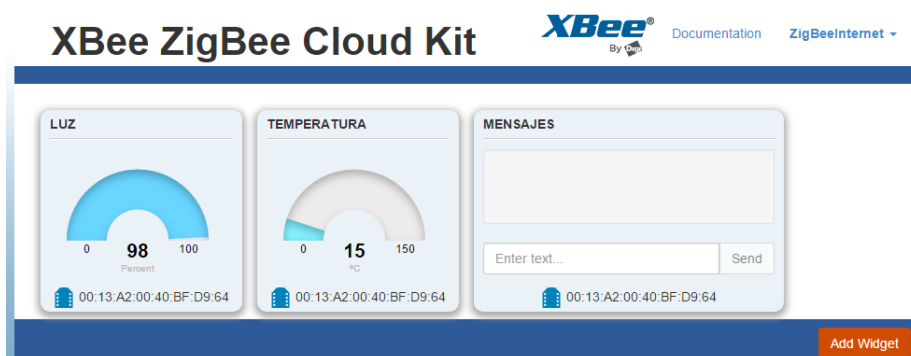


Figura 44: Pantalla de monitoreo del invernadero
Fuente: Elaboración Propia (11/04/2016)

8 Pruebas de funcionamiento del prototipo

Para probar el funcionamiento del prototipo se realizaron dos tipos de pruebas, las pruebas de laboratorio y pruebas de campo, las pruebas de laboratorio se realizaron en un cuarto con todos los dispositivos juntos para probar el correcto funcionamiento del prototipo. Las pruebas de campo se las realizaron en campo abierto y otra con obstáculos.

8.1 Pruebas de Laboratorio

Para las pruebas se conectó el prototipo dentro de un cuarto donde se introdujo el prototipo dentro del mini invernadero usando la energía de la casa y la conexión a Internet ISP.

- Se energiza el Gateway xbee comprobando que se hay registrado en nuestra red Zigbee, Wifi e Internet, esto se lo puede comprobar revisando los tres leds informativos del Gateway si se encuentran en color verde.

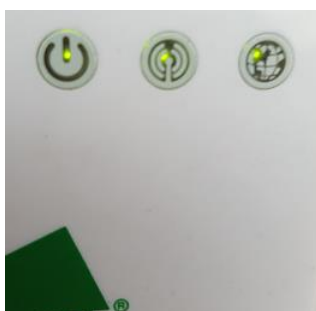


Figura 45: Leds informativos Gateway
Fuente: Elaboración Propia (24/04/2016)

- Se energiza el módulo xbee con un equipo laptop y se coloca dentro del mini invernadero.

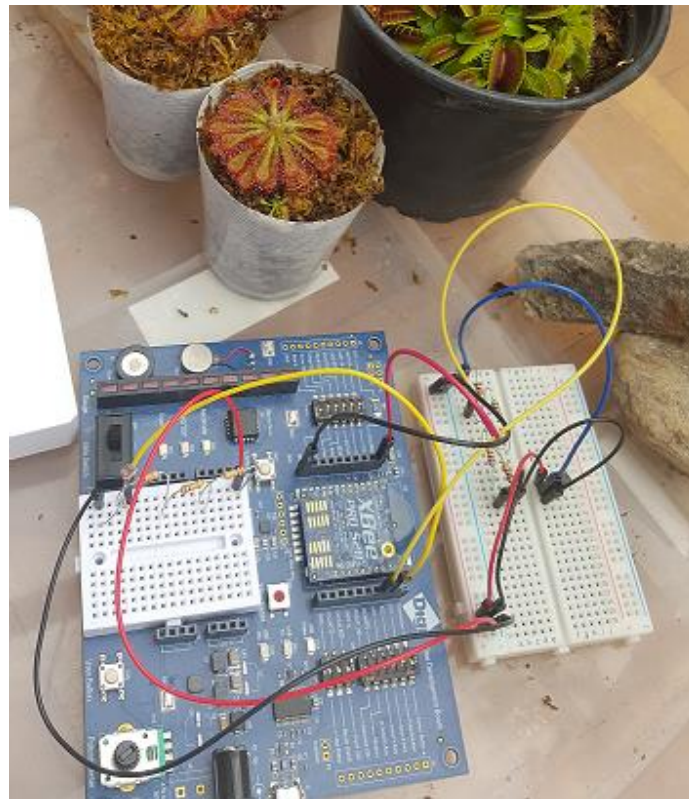


Figura 46: Prototipo dentro del invernadero
Fuente: Elaboración Propia (24/04/2016)

- Se comprueba el registro de datos de los sensores y el envío de texto por medio de un Browser.

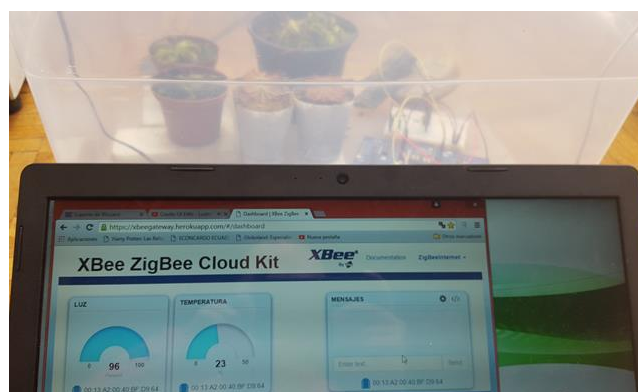


Figura 47: Browser mostrando los datos de sensores
Fuente: Elaboración Propia (24/04/2016)

Se utilizó el software XCTU ya que tiene un módulo de conexión serial para Xbee en donde es posible enviar y recibir texto.

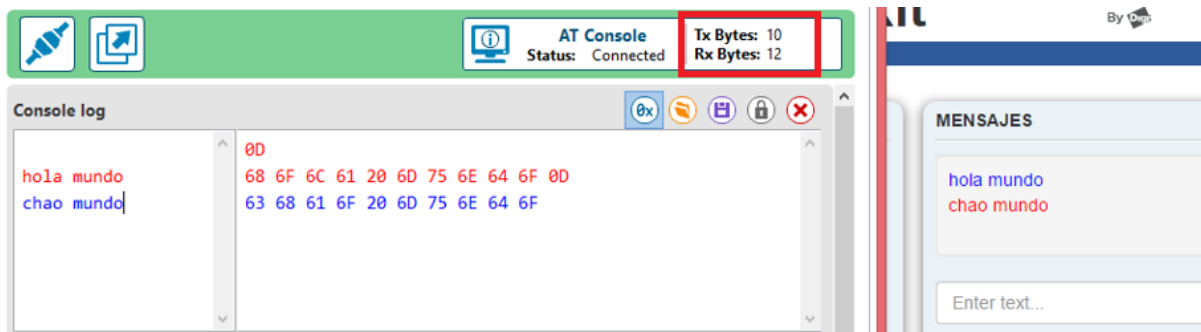


Figura 48: Envío de mensajes de Texto
Fuente: Elaboración Propia (24/04/2016)

Se puede apreciar en la consola la cantidad de bytes enviados y recibidos en este caso 10 bytes transmitidos y 12 bytes recibidos, se envió desde el Browser el texto “hola mundo” y se envió desde consola el texto “chao mundo”. También se puede observar que hay una luminosidad del 96% y una temperatura de 23 grados centígrados, esta medición se la hizo a las 3 de la tarde.

Con estas pruebas se entiende que la funcionalidad del prototipo es correcta como se ha planificado.

8.2 Pruebas de campo

Para este tipo de pruebas se considera dos escenarios, el primero se realizará pruebas de distancia sin línea de vista, es decir poner a prueba el prototipo en lugares con obstáculos como paredes, ventanas, modulares, etc. Y el segundo escenario se realizarán pruebas con línea de vista, es decir en campo abierto.

Para realizar las mediciones se utilizará el programa XCTU del mismo fabricante de los dispositivos Xbee, el cual posee un módulo de pruebas de distancia, enviando paquetes donde se puede parametrizar el tamaño del paquete, el tiempo de envío el tiempo de recepción y la cantidad de paquetes a enviar.

8.2.1 Pruebas con obstáculos

Para estas pruebas de transmisión de datos se realizara 4 muestreos a 10m, 20m, 30m y 50m enviando 100 paquetes, cada paquete se enviará por intervalos de 4000 ms con un tiemeout de recepción de 1000 ms (parámetros sugeridos por el fabricante).

Las pruebas realizadas a 10m sin línea de vista nos dan como resultado un 97% de paquetes transmitidos y 3 paquetes perdidos y una intensidad de señal de -41dbm.

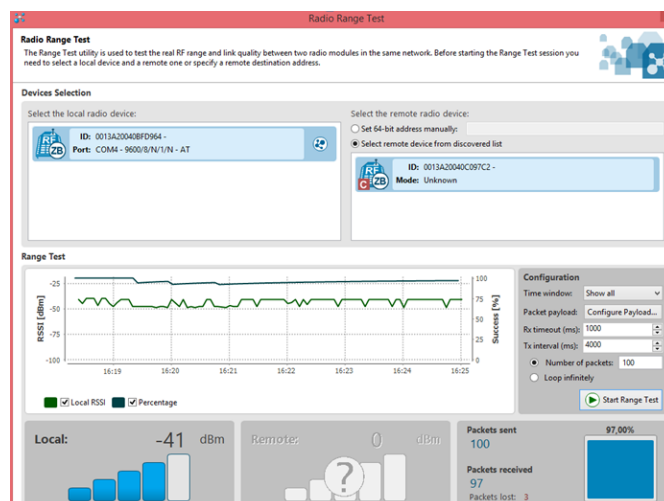


Figura 49: Resultado de las pruebas a 10m
Fuente: Elaboración Propia (24/04/2016)

Las pruebas realizadas a 20m sin línea de vista nos dan como resultado un 95% de paquetes transmitidos y 5 paquetes perdidos y una intensidad de señal de -76dbm.

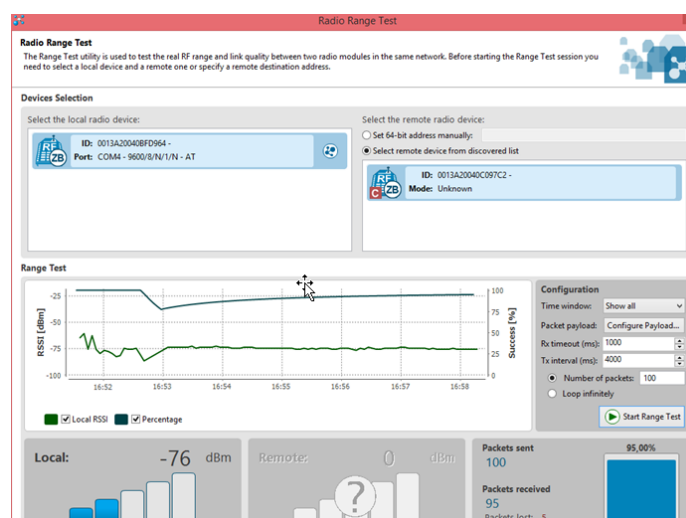


Figura 50: Resultado de las pruebas a 20m
Fuente: Elaboración Propia (24/04/2016)

Las pruebas realizadas a 30m sin línea de vista nos dan como resultado un 80% de paquetes transmitidos y 20 paquetes perdidos y una intensidad de señal de -84dbm.

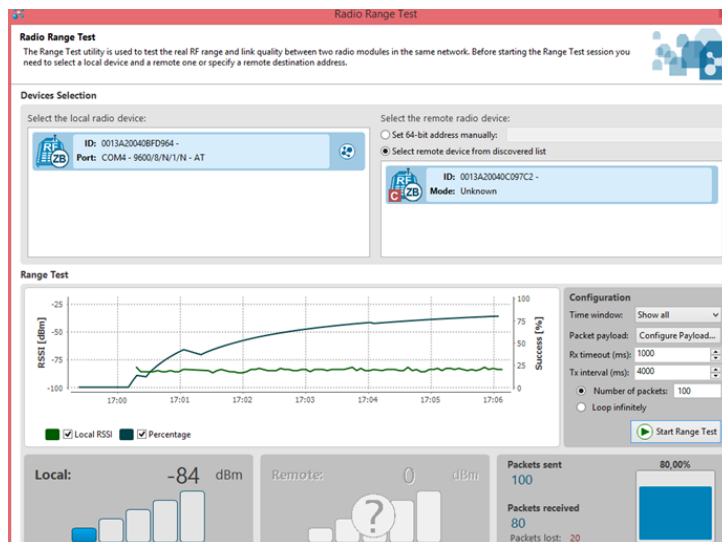


Figura 51: Resultado de las pruebas a 30m
Fuente: Elaboración Propia (24/04/2016)

Las pruebas realizadas a 50m sin línea de vista nos dan como resultado un 0% de paquetes transmitidos y 100 paquetes perdidos y una intensidad de señal de -84dbm.

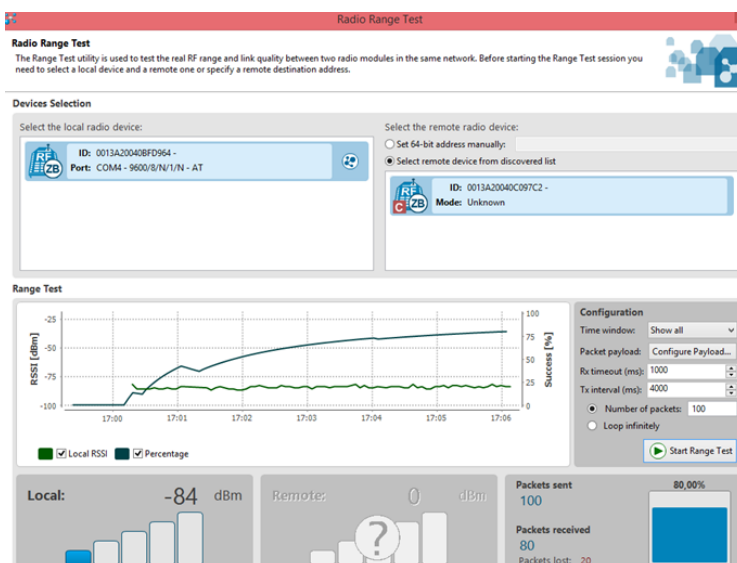


Figura 52: Resultado de las pruebas a 50m
Fuente: Elaboración Propia (24/04/2016)

En el siguiente cuadro podemos apreciar que las pruebas de hasta los 20 m de distancia con obstáculos se tiene un nivel de recepción y perdidas aceptable para el correcto funcionamiento del sistema de monitoreo, a partir de los 30m de distancia existe una pérdida importante de paquetes se puede apreciar que se tiene un %20 de perdidas, y a partir de los 50 m de distancia ya no se detecta nivel de señal siendo imposible el funcionamiento del sistema de monitoreo.

DISTANCIA	NIVEL DE SEÑAL	PAQUETES PERDIDOS	% DE EFECTIVIDAD
10m	- 41 dbm	3	%97
20m	- 76 dbm	5	%95
30m	- 84 dbm	20	%80
50m	Sin Señal	100	%0

Tabla 2: Resultado de Pruebas con Obstáculos
Fuente: Elaboración Propia (24/04/2016)

8.2.2 Pruebas con línea de Vista

Para estas pruebas se utilizará el Software XCTU para el análisis del rango de señal y la herramienta en GOOGLE MAPS para la referencia de las distancias para tomar las muestras. Las pruebas se realizarán en el parque Bicentenario en la ciudad de Quito y se realizarán 4 muestras, una a 50m, 100m, 150m, 200m.

Para energizar el Gateway en campo abierto se utilizó un arreglo de 4 baterías AA de 1,5 v y para energizar el módulo Xbee se utilizó un equipo Laptop.



Figura 53: Gateway energizada por Baterías
Fuente: Elaboración Propia (24/04/2016)

Las pruebas realizadas a 50m con línea de vista nos dan como resultado un 100% de paquetes transmitidos y 0 paquetes perdidos y una intensidad de señal de -66dbm.



Figura 54: Ubicación para las pruebas a 50m
Fuente: Elaboración Propia (24/04/2016)

Radio Range Test

The Range Test utility is used to test the real RF range and link quality between two radio modules in the same network. Before starting the Range Test session you need to select a local device and a remote one or specify a remote destination address.

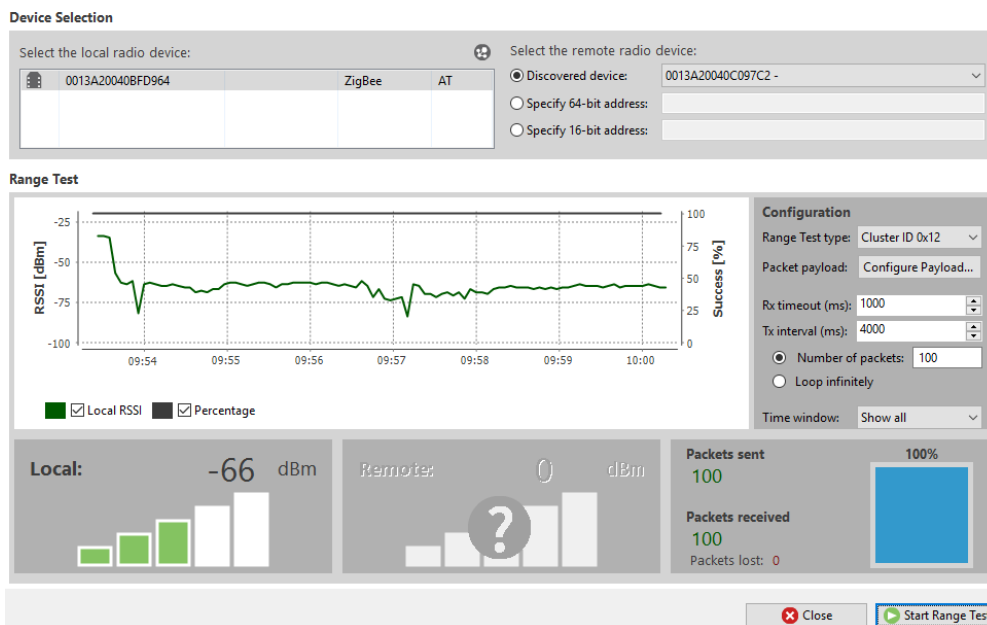


Figura 55: Resultado de las pruebas a 50m
Fuente: Elaboración Propia (24/04/2016)

Las pruebas realizadas a 100m con línea de vista nos dan como resultado un 100% de paquetes transmitidos y 0 paquetes perdidos y una intensidad de señal de -81dbm.



Figura 56: Ubicación para las pruebas a 100m
Fuente: Elaboración Propia (24/04/2016)

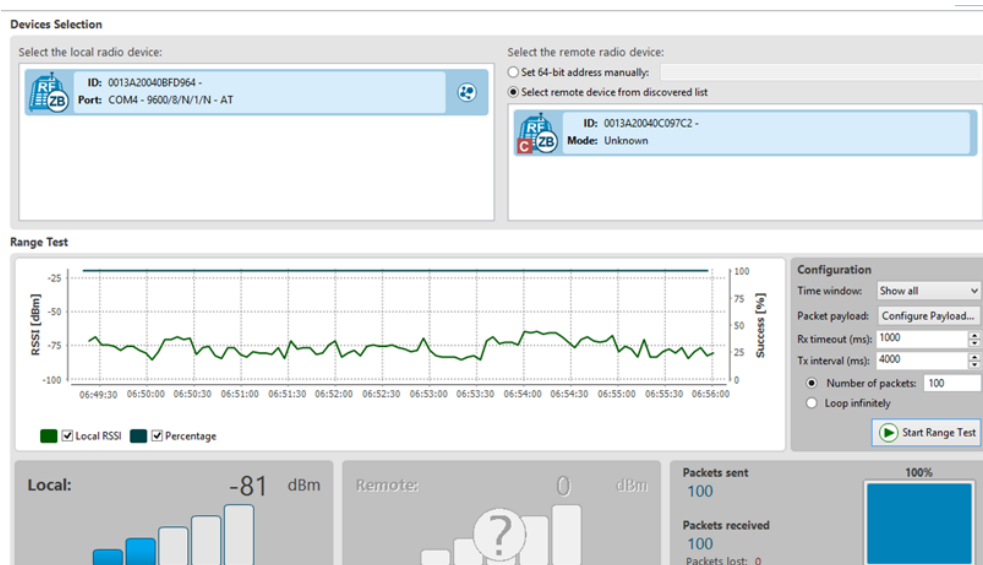


Figura 57: Ubicación para las pruebas a 100m
Fuente: Elaboración Propia (24/04/2016)

Las pruebas realizadas a 150m con línea de vista nos dan como resultado un 50% de paquetes transmitidos y 50 paquetes perdidos y una intensidad de señal de -81dbm.



Figura 58: Ubicación para las pruebas a 150m
Fuente: Elaboración Propia (24/04/2016)

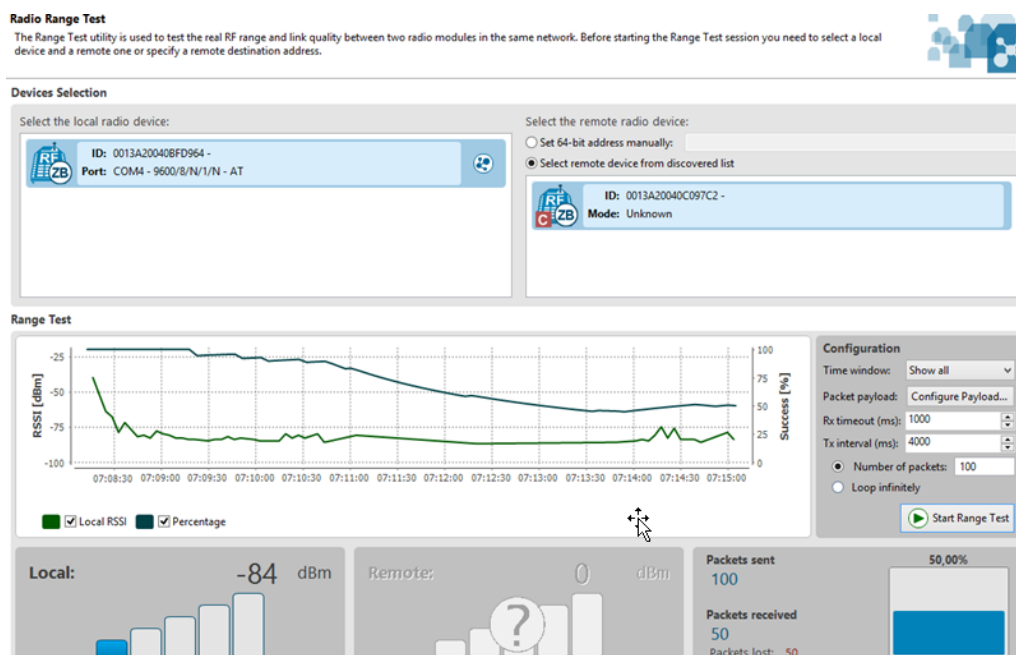


Figura 59: Resultado de las pruebas a 150m
Fuente: Elaboración Propia (24/04/2016)

Las pruebas realizadas a 200m con línea de vista nos dan como resultado un 0% de paquetes transmitidos y 100 paquetes perdidos y una intensidad de señal de -84dbm.



Figura 60: Ubicación para las pruebas a 200m
Fuente: Elaboración Propia (24/04/2016)

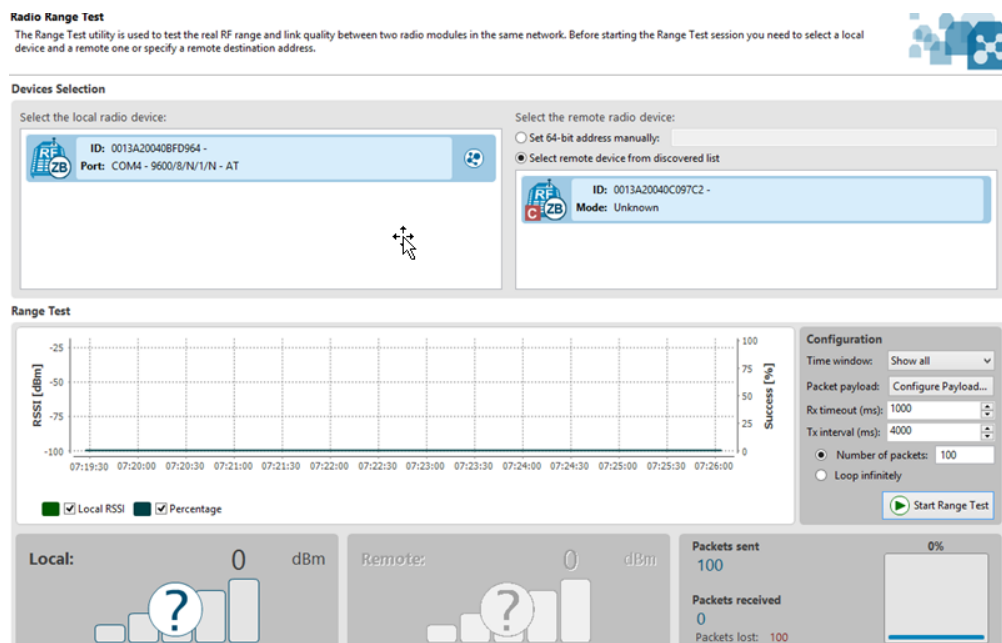


Figura 61: Resultado de las pruebas a 200m
Fuente: Elaboración Propia (24/04/2016)

En el siguiente cuadro podemos apreciar que las pruebas de hasta los 20 m de distancia con obstáculos se tiene un nivel de recepción y perdidas aceptable para el correcto funcionamiento del sistema de monitoreo, a partir de los 30m de distancia existe una pérdida importante de paquetes se puede apreciar que se tiene un %20 de perdidas, y a partir de los 50 m de distancia ya no se detecta nivel de señal siendo imposible el funcionamiento del sistema de monitoreo.

DISTANCIA	NIVEL DE SEÑAL	PAQUETES PERDIDOS	% DE EFECTIVIDAD
50m	- 66 dbm	0	%100
100m	- 81 dbm	0	%100
150m	- 84 dbm	50	%50
200m	Sin Señal	100	%0

Tabla 3: Resultado de Pruebas sin Obstáculos
Fuente: Elaboración Propia (24/04/2016)

9 Comparación del prototipo con otros sistemas de monitoreo

Haciendo una comparación con los sistemas de monitoreo antes mencionados se puede evidenciar que poseen características muy parecidas, todos por ejemplo pueden conectar aproximadamente la misma cantidad de sensores, todos usan software libre, todos publican sus datos en INTERNET, son fáciles de implementar, son de un precio accesible al usuario final, se acoplan al funcionamiento para varios propósitos, además hoy por hoy en casi todo lugar hay acceso a internet por lo que estos sistemas pueden utilizarse en cualquier momento y ante cualquier necesidad.

Al comparar las ventajas de tener un sistema de monitoreo que no solo utilice el INTERNET sino también las ventajas del estándar ZigBee se amplía aún más las prestaciones de un sistema para el monitoreo, ventajas como el bajo consumo energético que este estándar nos brinda, las grandes distancias que se pueden obtener incorporando la antena adecuada, se puede crear una red en malla y así aumentar aún más el área de cobertura sin dispositivos intermedios.

A continuación se describen las principales ventajas y desventajas de los sistemas de monitoreo mencionados con el prototipo de este caso de estudio:

PROYECTO	PRECIO UNITARIO	ESCALABILIDAD	NUMERO DE SENSORES	IMPLEMENTACIÓN	TECNOLOGIA	CONSUMO DE ENERGIA
Monitoreo Xbee	\$ 125	hasta 254 nodos por red ZigBee y hasta 257 clusters es decir se puede obtener suma red con 254 x 255= 64.770 nodos	3 entradas Sensores Digitales, 3 entradas para Sensores Analógicos es decir 6 sensores por módulos pudiéndose extender el sistema a 257 módulos Xbees es decir un total de 771 sensores en toda su capacidad	Fácil de implementar y configurar, los módulos xbee forman su red automáticamente, solo se necesita configuración si se quiere alguna configuración especial como tiempos de inactividad, tiempos de transferencia, etc.	inalámbrica (Tecnología ZigBee)	40 mA a 3.3V
AirPi	\$90 (No incluye Raspberry PI)	no es escalable	6 sensores análogos o digitales conectados simultáneamente	La placa Airpi es fácil de implementar que es plug and play hacia un RaspBerry Pi	alámbrica	5v, 900mA, aunque depende de la carga de trabajo de los 4 cores
Weather Meters	\$250(incluye arduino)	no es escalable	3 sensores simultáneamente	Se necesita conocimientos básicos de electrónica	alámbrica	5 v, 900 mA
GrovePi	\$90 (No incluye Raspberry PI)	no es escalable	15 sensores simultáneamente	Se necesita conocimientos básicos de electrónica	alámbrica	5v, 900mA, aunque depende de la carga de trabajo de los 4 cores

Tabla 4: Comparativo del Prototipo con los Sistemas de Monitoreo Ambiental
Fuente: Elaboración Propia (24/04/2016)

9.1 Ventajas del prototipo de monitoreo ambiental con otros sistemas de monitoreo

Basándonos en el anterior cuadro comparativo se pueden observar las siguientes ventajas:

➤ Fácil de escalar

Con este sistema de monitoreo ambiental si por ejemplo se necesita incorporar otro módulo Xbee ya sea para cubrir más distancias con más sensores pues simplemente se realiza las diferentes configuraciones para incorporarse a la red ZigBee y está listo, se cubrirá un rango más amplio de monitoreo.

➤ Es un sistema Integral

Los sistemas de monitoreo antes mencionados trabajan de forma individual y si se incorpora otra unidad de monitoreo pues la tendríamos que ver individualmente es decir no se integran entre ellos, por el contrario del sistema de monitoreo propuesto, si se incorpora otro módulo Xbee ya sea con los mismos sensores o con otro tipo los datos que tendremos de los módulos Xbee se observarán en el mismo Dashboard es decir todos los datos de los sensores se los observará en la misma ventana del Browser.

➤ Mayor área de cobertura

Con los estándares WiFi y Ethernet se puede cubrir largas distancias con Ethernet 100m y con Wifi hasta 500 sin equipos intermedios, pero con el estándar ZigBee se puede alcanzar hasta 1.6 km utilizando la antena y la potencia máxima de transmisión, esto de nodo a nodo y sin equipos intermedios, es decir podemos utilizar 2 módulos Xbee en una distancia de 3.2 km hasta el GateWay.

➤ Mayor tiempo de vida

Con una configuración adecuada del de la propiedad de *Sleep* de los módulos Xbee y según las necesidades de transmisión de datos con baterías de 40 mA a 3.3V se puede energizar varios meses y incluso un par de años sin recargas.

9.2 Desventajas del prototipo de monitoreo ambiental con otros sistemas de monitoreo

➤ Altas tasas de transferencia de datos

El estándar 802.11g nos permite una velocidad de 54 Mbps y el estándar 802.3 nos permite una velocidad de 1000 Mbps lo cual es una altísima tasa de transferencia comparada a los 250 Kbps de ZigBee esta es la única desventaja que puede evidenciarse ante los anteriores sistemas.

10. Conclusiones y Recomendaciones

10.1 Conclusiones

- Este sistema de monitoreo sirve tanto para invernaderos dentro de casa o llamados “Indoors” o a campo abiertos, dentro de casa con una distancia máxima de 30 metros de Gateway y 150 metros en exteriores esto utilizando 1 solo módulo Xbee, para cubrir más área de monitoreo es necesario implementar más módulos Xbee para cubrir las necesidades requeridas.
- Con un control del ambiente se puede observar que las plantas crecen más rápido y sanas, también controlando la temperatura y luminosidad se optimiza recursos como el riego y cuando el cultivador decide cuando quiere cambiar de etapa o de ciclo de la planta, es decir si la quiere mantener en ciclo vegetativo, después disminuir la cantidad de luz y obligar a florecer a las plantas, y bajando la temperatura para que las plantas hibernen.
- Los datos obtenidos con el prototipo sirven tanto para invernaderos caseros como profesionales, la precisión de los datos solo depende del tipo de sensor y su sensibilidad más que del propio prototipo.
- Comparado con los sistemas de monitoreo anteriores la ventaja de usar módulos Xbee es el área de cobertura que se puede lograr vinculando más módulos Xbee en un mismo sistema de control además que el consumo energético es mucho menor.

10.2 Recomendaciones

- Para mejorar las prestaciones de este sistema se recomienda aumentar la cantidad de sensores para tener mayor control de las variables climáticas, como también se puede implementar una base de datos para llevar un control histórico y estadístico del control del invernadero.
- Para alcanzar mayores distancias se debe incorporar al módulo Xbee una antena adecuada y así evitar la pérdida de paquetes al aumentar la distancia.
- Si es necesario aumentar la duración de baterías de los módulos Xbee se debe modificar configuración del parámetro *Sleep* y así por ejemplo que quede el módulo dormido por 1 minuto y transmita datos por 10 segundos y después vuelva a dormir.

- Si se quiere obtener una respuesta rápida de la entrega de información de los sensores para visualizarlos en el Browser basta con configurar los tiempos de muestreo, en nuestro caso es cada 5 segundos o 5000 milisegundos que es la unidad que maneja el módulo Xbee.
- En la búsqueda de módulos Xbee en Ecuador no se encontró información sobre ningún distribuidor para el país por lo que hace suponer que la implementación de esta tecnología es una buena oportunidad para realizar proyectos emprendedores como existen en otros países, proyectos como la lectura remota de medidores de agua y consumo eléctrico en los hogares a través de internet obteniendo datos de los módulos Xbee, la supervisión de nuestros volcanes sin tener que estar en la zona de peligro, etc.

11 Bibliografía

AirPi. (14 de Diciembre de 2015). *AirPi*. Obtenido de <http://airpi.es/>

Alibaba. (25 de Noviembre de 2015). *Alibaba*. Obtenido de http://rexense.en.alibaba.com/product/501808351212951716/ZigBee_WiFi_Wireless_Gateway_for_Smart_Home_Automation_HA_compliant.html

ARDUINO. (Noviembre de 2015). *ARDUINO*. Obtenido de <https://www.arduino.cc/en/Main/ArduinoWirelessProtoShield>

Brultech. (10 de Enero de 2016). *Brultech*. Obtenido de http://www.brultech.com/home/store/product.php?id_product=22

Cisco. (20 de Enero de 2016). *Cisco*. Obtenido de <https://cisco.com>

Corporacion Universitaria Autonoma del Cauca . (s.f.). *EJEMPLARIZACION DE COMUNICACIÓN ENTRE DOS MODUULOS* . Colombia: Corporacion Universitaria Autonoma del Cauca .

debian. (2016). *debian*. Obtenido de <https://packages.debian.org/search?keywords=lucvview>

Dexter Industries. (2016). *Dexter Industries*. Obtenido de <http://www.dexterindustries.com/shop/grovepi-starter-kit-raspberry-pi/>

Digi. (20 de Octubre de 2015). *Digi*. Obtenido de <http://www.digi.com>

Duarte, A. (8 de Noviembre de 2015). *Arduino y Xbee*. Obtenido de <http://www.andresduarte.com/arduino-y-xbee>

ESPE. (2 de Febrero de 2016). *Repositorio Digital ESPE*. Obtenido de <http://repositorio.espe.edu.ec/handle/21000/6443/statistics>

Fritzing. (17 de Octubre de 2015). *Fritzing*. Obtenido de <http://fritzing.org/home/>

Harman. (19 de Septiembre de 2015). *AMX by HARMAN*. Obtenido de <http://www.amx.com/products/NXR-ZGW.asp>

Heroku. (17 de Febrero de 2016). *Heroku*. Obtenido de <https://toolbelt.heroku.com/>

Lorenzo, P. (2012). El cultivo en invernadero y su relación con el clima. *Cuadernos de estudios Agroalimentarios*, 23-44.

Meléndez, F., & Cano, E. (2015). PATHFI: Sistema portátil de monitoreo de temperatura, humedad relativa, presión atmosférica y altitud utilizando el protocolo Wi-Fi. Oaxaca, México: Universidad de Itsmo.

Putty. (20 de Octubre de 2015). *putty*. Obtenido de <http://www.putty.org/>

Raspberry Pi Foundation. (4 de Diciembre de 2015). *Raspberry Pi Foundation*. Obtenido de <https://www.raspberrypi.org/>

Sparkfun. (Septiembre de 2015). *Sparkfun*. Obtenido de <https://www.sparkfun.com/products/8942>

Telegesis. (2016). *Telegesis*. Obtenido de <http://www.telegesis.com/products/zigbee-gateway-consumer-access-device/>

Texas Instruments. (13 de Octubre de 2015). *Texas Instruments*. Obtenido de <http://www.ti.com/tool/CC2531EM-IOT-HOME-GATEWAY-RD>

West Monroe. (24 de Septiembre de 2015). *West Monroe*. Obtenido de <http://www.westmonroepartners.com/>

12 ANEXOS

ANEXO 1

Código Fuente para la programación y lectura de datos desde los Widgets

Código JavaScript para la obtención de datos de la fotocelda:

Fuente: www.digi.com

```
/*
 * This Source Code Form is subject to the terms of the Mozilla Public License,
 * v. 2.0. If a copy of the MPL was not distributed with this file, You can
 * obtain one at http://mozilla.org/MPL/2.0/.
 *
 * Copyright (c) 2014 Digi International Inc., All Rights Reserved.
 */

'use strict';

angular.module('XBeeGatewayApp')
  .directive('gaugeWidget', function (widgetRegistry, utils) {
    // called after DOM element is compiled
    var linker = function postLink(scope, element) {
      scope.$element = element;
      var type = 'gauge';
      var spec = widgetRegistry.get(type);

      // See http://lodash.com/docs#bind
      // (dataUpdate simply calls scope.updateHandler)
      var dataUpdate = _.bind(scope.updateHandler, scope);
      utils.postlinkWidget(type, scope, element, spec, dataUpdate);

      scope.value = undefined;
    };

    // AngularJS directive setup
```

```

return {
    templateUrl: "widgets/gaugeWidget/gaugeWidget.tpl.html",
    restrict: 'AE',
    link: linker,
    controller: 'simpleDataHolder',
    scope: { widget: "=gaugeWidget" }
};
})

// This function, referred to in AngularJS as a "run block", is called by
// AngularJS after the injector is created and is used to bootstrap the
// application. The XBee ZigBee Cloud Kit makes use of the run block
// to add widget definitions to the widget registry at start-up.
.run(function(widgetRegistry) {
    // Adding the widget to the widget registry
    var widget_type_key = 'gauge';
    var widget_description = 'Gauge Widget';
    var widget_spec = {
        // Whether or not the widget is built-in or user-created
        // (i.e., whether the code is in /src/app or /src/common)
        builtin: true,
        // widget size: X,Y (columns, rows)
        size: [2, 2],
        // description appearing in 'Widget Type' list when adding new
        // widgets
        description: widget_description,
        directive: "gauge-widget",
        // camel-case version of directive
        directive_c: "gaugeWidget",

        // properties pertaining to widget settings
        /*
        has_input: does the widget's data get updated from Device Cloud?
        sends_output: does the widget send data to the device?
        input_xform: can the user specify a custom transformation to apply

```

```

        to incoming data? (optional)
options: list of objects defining the settings associated with this
        widget type
    - options generally follow the Revalidator API
      (http://github.com/flatiron/revalidator)
    */
    has_input: true,
    sends_output: false,
    input_xform: true,
    options: [
      {key: "units", type: "string", label: "Units", required: false,
        "default": ""},
      {key: "low", type: "number", label: "Low value", required: false,
        "default": 0},
      {key: "high", type: "number", label: "High value",
        required: false, "default": 2500,
        dependencies: 'low', conform: function (val, obj) {
          var low = obj.low;
          return low < val;
        }},
      messages: {conform: "High value must be greater than low value"}}
    ]
  };

  // DO NOT CHANGE ANY CODE BELOW HERE.
  widgetRegistry.put(widget_type_key, widget_spec);
});

```

Código JavaScript para la obtención de datos del sensor de temperatura:

Fuente: www.digi.com

```

/*
* This Source Code Form is subject to the terms of the Mozilla Public License,

```

```
* v. 2.0. If a copy of the MPL was not distributed with this file, You can
* obtain one at http://mozilla.org/MPL/2.0/.
*
* Copyright (c) 2014 Digi International Inc., All Rights Reserved.
*/
```

```
'use strict';
```

```
angular.module('XBeeGatewayApp')
.directive('gaugeWidget', function (widgetRegistry, utils) {
  // called after DOM element is compiled
  var linker = function postLink(scope, element) {
    scope.$element = element;
    var type = 'gauge';
    var spec = widgetRegistry.get(type);

    // See http://lodash.com/docs#bind
    // (dataUpdate simply calls scope.updateHandler)
    var dataUpdate = _.bind(scope.updateHandler, scope);
    utils.postlinkWidget(type, scope, element, spec, dataUpdate);

    scope.value = undefined;
  };

  // AngularJS directive setup
  return {
    templateUrl: "widgets/gaugeWidget/gaugeWidget.tpl.html",
    restrict: 'AE',
    link: linker,
    controller: 'simpleDataHolder',
    scope: { widget: "=gaugeWidget" }
  };
})

// This function, referred to in AngularJS as a "run block", is called by
```

```

// AngularJS after the injector is created and is used to bootstrap the
// application. The XBee ZigBee Cloud Kit makes use of the run block
// to add widget definitions to the widget registry at start-up.
.run(function(widgetRegistry) {
    // Adding the widget to the widget registry
    var widget_type_key = 'gauge';
    var widget_description = 'Gauge Widget';
    var widget_spec = {
        // Whether or not the widget is built-in or user-created
        // (i.e., whether the code is in /src/app or /src/common)
        builtin: true,
        // widget size: X,Y (columns, rows)
        size: [2, 2],
        // description appearing in 'Widget Type' list when adding new
        // widgets
        description: widget_description,
        directive: "gauge-widget",
        // camel-case version of directive
        directive_c: "gaugeWidget",

        // properties pertaining to widget settings
        /*
        has_input: does the widget's data get updated from Device Cloud?
        sends_output: does the widget send data to the device?
        input_xform: can the user specify a custom transformation to apply
                     to incoming data? (optional)
        options: list of objects defining the settings associated with this
                 widget type
                 - options generally follow the Revalidator API
                   (http://github.com/flatiron/revalidator)
        */
        has_input: true,
        sends_output: false,
        input_xform: true,

```

```

options: [
  {key: "units", type: "string", label: "Units", required: false,
    "default": ""},
  {key: "low", type: "number", label: "Low value", required: false,
    "default": 0},
  {key: "high", type: "number", label: "High value",
    required: false, "default": 2500,
    dependencies: 'low', conform: function (val, obj) {
      var low = obj.low;
      return low < val;
    },
    messages: {conform: "High value must be greater than low value"}}
]
};

// DO NOT CHANGE ANY CODE BELOW HERE.
widgetRegistry.put(widget_type_key, widget_spec);
});

```

Código Java Scrip para la implementación del módulo del chat

Fuente: www.digi.com

```

/*
 * This Source Code Form is subject to the terms of the Mozilla Public License,
 * v. 2.0. If a copy of the MPL was not distributed with this file, You can
 * obtain one at http://mozilla.org/MPL/2.0/.
 *
 * Copyright (c) 2015 Digi International Inc., All Rights Reserved.
 */

'use strict';

angular.module('XBeeGatewayApp')

```

```

//jshint -W098
.controller('serialWidgetCtrl', function ($scope, utils, $log, dashboardApi, notificationService) {
    $scope.data_sending = false;
    $scope.last_received_timestamp = undefined;

    $scope.getStreamUpdateHandler = function (newData) {
        // For the serial widget, data *should* come in as an object of the form
        // {"content": "...", "format": "base64"}
        var _newData = newData.value;
        if (!_isEmpty(_newData)) {
            $log.warn("Serial widget got malformed data", newData);
            return;
        }
        var _timestamp = newData.timestamp;
        newData = _newData;

        // XBee Gateway
        if (angular.isString(newData)) {
            var decoded = utils.base64_decode(newData);

            if (!_isEmpty(decoded)) {
                // Might not be decoded.
                newText = newData;
            } else {
                // Successfully decoded the data.
                newText = decoded;
            }
            $log.info("Adding text:", newText);
            $scope.displaySerialText(newText, true);
            $scope.last_received_timestamp = _timestamp;
            return;
        }
        // Or, XBee Wi-Fi
        else if (newData.content !== undefined){

```

```

    // Check format to see if we should b64 decode
    var newText;
    if(newData.format === "base64"){
        newText = utils.base64_decode(newData.content);
    } else {
        newText = newData.content;
    }
    $scope.displaySerialText(newText, true);
    $scope.last_received_timestamp = _timestamp;
} else {
    // TODO display some error?
    $log.warn('Serial widget got malformed data', newData);
}
}

$scope.setStreamUpdateHandler = function (newData) {
    // Updates to the set stream could be handled here too, as we use the same for get/set
    return;
}

$scope.serialEnterKeypress = function($event){
    if($scope.serialOutText){
        $scope.sendText($scope.serialOutText);
    }
    $event.preventDefault();
}

$scope.sendText = function(text) {
    $scope.data_sending = true;

    if ($scope.widget.add_carriage_returns) {
        // Insert a CR before/after string to make it show on new line
        // on both ends
        var cr = String.fromCharCode(13);

```



```

        text = cr + text + cr;
    }

    dashboardApi.send_serial($scope.widget.device, $scope.widget.radio, text).then(
        function(result){
            // On success, show the sent text
            $scope.displaySerialText(text, false);
            // Clear the input box for next entry
            $scope.serialOutText = null;
            // Reenable input
            $scope.data_sending = false;
        },
        function(reason){
            notificationService.error("Error sending text. Please try again.");
            // Reenable input
            $scope.data_sending = false;
        });
    }

    $scope.$watch('data_sending', function (sending) {
        if (sending) {
            $scope.widgetState = 1;
        } else {
            $scope.widgetState = 0;
        }
    });
})

.directive('serialWidget', function (widgetRegistry, utils, dataStreams, $log) {
    // called after DOM element is compiled
    var linker = function postLink(scope, element) {
        scope.$element = element;
        var type = 'serial';
        var spec = widgetRegistry.get(type);
    }

```

```

// See http://lodash.com/docs#bind
// (dataUpdate simply calls scope.updateHandler)
var getCallback = _.bind(scope.getStreamUpdateHandler, scope);
var setCallback = _.bind(scope.setStreamUpdateHandler, scope);
utils.postlinkWidget(type, scope, element, spec, getCallback, setCallback);
// Any more fancy footwork can be done here.

// Manually listen for serialIn
var device = scope.widget.device;
var inputStream = "xbee.serialIn/[" + scope.widget.radio + "]!";
$log.debug("Listening for serial stream: ", device, inputStream);
var removeListener = dataStreams.listen(device, inputStream,
scope.getStreamUpdateHandler);
scope.$on('$destroy', function () {
    removeListener();
});

// Widget display area
// Use jquery to find, Angular's jqLite doesn't support selector
var output_pane = $(element).find(".serial-display");

scope.displaySerialText = function(text, isInbound){
    var $newText = null;
    if(isInbound){
        $newText = $('<span/>').addClass("serial-in");
        // Show incoming text inline
        // Split string to handle any Carriage Returns
        if(text === "\r"){
            $newText.append($('<br>'));
        } else {
            var snippets = text.split("\r");
            _.each(snippets, function (snippet) {
                // CR at start & end will make cause empty strings
                if(snippet === ""){

```

```

        $newText.append($('<br>'));
    } else {
        $newText.append($('<span/>').text(snippet));
    }
    });
}
} else {
    // Outgoing should be in it's own line
    $newText = $('<p/>').text(text).addClass("serial-out");
}
$(output_pane).append($newText);
$(output_pane).scrollTop($(output_pane)[0].scrollHeight);
}
};

// AngularJS directive setup
return {
    templateUrl: "widgets/serialWidget/serialWidget.tpl.html",
    restrict: 'AE',
    link: linker,
    controller: 'serialWidgetCtrl',
    scope: { widget: "=serialWidget", widgetState: "=state" }
};
})

// This function, referred to in AngularJS as a "run block", is called by
// AngularJS after the injector is created and is used to bootstrap the
// application. The XBee ZigBee Cloud Kit makes use of the run block
// to add widget definitions to the widget registry at start-up.
.run(function(widgetRegistry) {
    // Adding the widget to the widget registry
    var widget_type_key = 'serial';
    var widget_description = 'Serial Data Widget';
    var widget_spec = {
        // Whether or not the widget is built-in or user-created

```

```

// (i.e., whether the code is in /src/app or /src/common)
builtin: true,
// widget size: X,Y (columns, rows)
size: [3, 2],
// description appearing in 'Widget Type' list when adding new
// widgets
description: widget_description,
directive: "serial-widget",
// camel-case version of directive
directive_c: "serialWidget",

// properties pertaining to widget settings
/*
has_input: does the widget's data get updated from Device Cloud?
sends_output: does the widget send data to the device?
input_xform: can the user specify a custom transformation to apply
               to incoming data? (optional)
options: list of objects defining the settings associated with this
          widget type
          - options generally follow the Revalidator API
            (http://github.com/flatiron/revalidator)
*/
has_input: false,
sends_output: false,
options: [
  {key: "add_carriage_returns", label: "Add Carriage Returns",
    type: "boolean", required: false, 'default': true}
]
};

// DO NOT CHANGE ANY CODE BELOW HERE.
widgetRegistry.put(widget_type_key, widget_spec);
});

```